Basic python practice

1. If this is your first time encountering dictionaries, running the following lines of code (one at a time) and trying to understand the results may help you get your footing before moving onto the remaining exercises. If you are confident using dictionaries, you can skip to exercise 2. (If you are getting an ‘invalid character’ error after copying and pasting, try typing the code into the console manually)
    a. my_dict = {‘common name’: ‘barn owl’, ‘genus’: ‘Tyto’, ‘species’: ‘alba’}
    b. my_dict
    c. dir(my_dict)
    d. my_dict.keys()
    e. my_dict.values()
    f. my_dict.items()
    g. my_dict[‘common name’]
    h. my_dict.update({‘max speed (km/h)’: 32}
    i. my_dict[‘max speed (km/h)’]
    j. my_dict[‘weight (kg)’] = 0.6
    k. my_dict[‘weight (kg)’]
    l. my_dict[‘weight (kg)’] += 0.01
    m. my_dict[‘weight (kg)’]
    n.
    ```
    for key, val in my_dict.items():
        print(‘{}: {}’.format(key, val))
    ```

2. Create a function that calculates the number of A’s, C’s, G’s, and T’s in a DNA sequence.
    a. Do this by looping through the string while keeping track of the number of nucleotides using a dictionary, using each nucleotide as a key, and the number of nucleotides as a value. Print out the number of each nucleotide in a pretty way. Identify the most common and least common nucleotide.

       For the next two problems you might find this page useful: https://docs.python.org/3/library/collections.html
    b. For situations like this, a default dictionary is often more useful than normal dictionary. One of the issues with part a is that you have to initialize the value as 0. A default dictionary can automatically do that for you. The defaultdict is a subclass of dictionary that is found in the collections module. Implement your code using the defaultdict.
    c. Finally use a Counter object from the collections module. Again print out the number of each nucleotide in a pretty way and print out the most common and least common nucleotide.

3. Create a dictionary:
   student = {"firstname": "Tyler",   "lastname": "Smith", "homework": [82.0, 0.0, 87.0, 75.0, 22.0],   "quizzes": [93.0, 0.0, 75.0, 78.0], "tests": [100.0, 100.0] }

   Create a function that prints out the name of the student followed by the overall grade in the class. To calculate the overall grade, first throw out the lowest quiz grade and the lowest homework grade. Then use the following weights to calculate an overall grade: 25% from homework, 25% from quizzes, and 50% from tests.

4. The os module is useful for file directory type operations. But as with anything involving file manipulation, be careful! You don't want to accidentally delete something important. Reference the os module documentation if you get stuck (https://docs.python.org/3/library/os.html). Figure out how to use os to:
   a. Print the current directory you are in
   b. Create a directory called TempDir
   c. Delete a directory called TempDir
   d. Print all of the files in your home directory
   e. Print all of the directories in your home directory
   f. Use an os function combined with a list comprehension to print out all of the files in your home directory that start with a '.'

5. You have a list of names of patients that are participating in a trial on whether ivermectin is effective at curing covid-19. You need to randomly assign them to one of two groups, the treatment group (that gets ivermectin) and the control group (that gets a placebo). Reference the python documentation for the random module (https://docs.python.org/3/library/random.html) if you get stuck. Do this in two ways:
   a. Use a method of the random module to randomize the order of the list and assign the first half to one group and the second half to another group
   b. Use a method of the random module to return a 0 or 1 and use that for each person to assign them to one of two groups