PGR107 – Python Programming
Spring 2023
Sets & Dictionaries

1. Given the set definitions below, answer the following questions:
    set1 = {1, 2, 3, 4, 5}
    set2 = {2, 4, 6, 8}
    set3 = {1, 5, 9, 13, 17}

    a. Is set1 a subset of set2?
    b. Is the intersection of set1 and set3 empty?
    c. What is the result of performing set union on set1 and set2?
    d. What is the result of performing set intersection on set2 and set3?
    e. What is the result of performing the set difference on set1 and set2 (set1 – set2)?

2. Given a dictionary
    grade_count = {"A": 8, "D": 3, "B": 15, "F": 2, "C": 6}

    write the python statement(s) to print:
    a. All the keys.
    b. All the values.
    c. All the key and value pairs.
    d. All the key and value pairs in key order.
    e. The average value.
    f. A chart similar to the following in which each row contains a key followed by a number of asterisks equal to the key's data value. The rows should be printed in key order, as shown below.

        A: ********
        B: ***************
        C: ******
        D: ***
        F: **

3. Write a program using dictionary that translates phone numbers in digits (entered by the user) to words.

    Sample run:

    Phone Number: 1234

    One Two Three Four

4. Write a program that keeps a dictionary in which both keys and values are strings—names of students and their course grades. Prompt the user of the program to add or remove students, to modify grades, or to print all grades. The printout should be sorted by name formatted like this:

    Carl: B+
    Francine: A
    Joe: C
    Sarah: A

5. Consider a dictionary of different usernames (keys) and passwords (values). Write a function called **accept_login** () with three parameters: the dictionary, a username, and a password. The function should return **True** if the user exists and the password is correct, then the message "**Login Successful**" is printed. Otherwise, the function returns **False**, then the message "**Login Failed**" is printed.

6. Write a function **def one_to_one (d)** that takes a dictionary *d* and returns **True** if every value in *d* has only one corresponding key. For example, if d = {'a': 4, 'b': 5, 'c': 3}, the function returns **Ture** but if d = {'a': 2, 'b': 4, 'c': 2}, the function returns **False**.