ENGR 101 Introduction to Programming
Study Questions - Week 10

1. What is the result of the following codes?

   a = [4, 2, 8, 6, 5]
   b = a
   b[3]=99
   print **a**

   ---

   a = [21, 22, 23]
   b = [21, 22, 23]
   print **a == b**
   print **a is b**

   b = a
   print **a == b**
   print **a is b**

   b[0] = 5
   print **a**

2. Let's say we have a dictionary that has strings as its keys and integers as its values. Write a function that will sum up the values of this dictionary.

3. Write a function that takes 2 lists of the same length, and assigns the elements of the first list as the dictionary keys, and the elements of the second list as the values of that dictionary. See what happens if the first list has duplicate elements.

4. Write a function that initializes a dictionary. The keys of this dictionary will be between 2 and 12 (including 2 and 12), and the values are square of the keys.

5. Write a function that takes a list and a dictionary as input. It should check if the elements of the list exist as values in the dictionary. If so, then return those values as a list.

6. Write a function which will take a dictionary and a number as arguments. It will search through dictionary's values, and return True if that number appears in the dictionary as an integer or as a string.

7. We have a list of shopping items that we have to buy. After we finish shopping, we created a dictionary that has items bought as its keys and their prices as its values. Your program should delete the items that we bought from the shopping_list, and calculate the total price.

```
>>>shopping_list=["apples","bananas","kiwi","pears","tomatoes"]
>>>shopping_done={"apples":3,"kiwi":4,"tomatoes":1}
['bananas', 'pears']
8
```

8. You have a dictionary that will look like this: class_1={"John":3,"Steve":5,"Emma":4} where student name and their grades are mapped in it as keys and values, respectively. Calculate the average class grade.

9. Write a function which takes a dictionary, user name, and password (users, username, password) as its arguments. It verifies if that person exists in our database, meaning if that pair exists in our users dictionary.

```
>>>users={"user1":"pasword1","user2":"pasword2","user3":"pasword3"}
>>>print check_pass(users,"user1","password2")
Could not log in
```

10. Write a function that takes as arguments a dictionary and a variable. The function should return a list that contains the keys that map to the value of the variable in the dictionary.

11. For a given list of numbers, write a function that returns a list of numbers that are greater than the average.

12. Define a function that takes a list as input and returns the list sorted in reverse order (descending).

13. Define a function that takes two lists as inputs. The first list is a data vector containing missing values represented by *'NA'* string and the second list has number of elements equal to number of 'NA's in the first list. The function should substitute all 'NA's in the first list with values from the second list (taken in order from beginning of list).

14. Imagine two lists (integer values) of equal length where the first one contains powers of x and the second list contains co-efficients of powers of x. Define a function ***calc_polynomial*** that takes these two lists as well as a value for x and returns the value of polynomial defined by the two lists.

```
>> calc_polynomial([3, 2, 1], [2, 0, 4], 3)        #evaluating 2x^3 + 4x at x = 3
66
```

15. Define a function that takes a list as input. The list is a sample space for a probability experiment. The function returns a dictionary whose keys are distinct elements from the list and values are probabilities for corresponding keys.

16. Imagine a dictionary with the structure *{student_id:[student_major, student_cgpa]}*. Assume majors are represented by strings like 'CS', 'EE', etc. Now define a function ***average_cgpa_by_major*** that takes such a dictionary and a string for major as inputs returns the average cgpa for all students with that major.

17. Recall Question 16. Define a function ***extend_student_db*** that takes the dictionary in question 15 as well as another dictionary with structure {student_id:student_name} as two inputs and adds student_name field to the list of values in the first dictionary. The new dictionary will have structure *{student_id:[student_name, student_major, student_cgpa]}*

18. Imagine a dictionary with the structure *{name:height}*. Define a function ***height_filter*** that takes such a dictionary and a number h as inputs and returns a list containing names of people with height greater than h.

19. Recall Question 18. Define a function ***height_reverse_lookup*** that takes the dictionary from question 18 as input and returns its reverse dictionary i.e. the keys will be heights and values will be names (or list of names in case multiple people have the same height).

20. Define a function that takes a number n as input and returns a dictionary containing the first n Fibonacci numbers. The keys of the dictionary are integers from 1 to n and the values are corresponding n^{th} Fibonacci numbers.

    Note: Do NOT define a function for Fibonacci numbers. Use your dictionary to grow it.

21. Following code finds the mode (most frequent value) of height in the {name:height} dictionary. Fill in the missing lines.

```
height_freq = dict()
for key in name_height_dict:
    value = name_height_dict[key]
    if value not in height_freq:
        height_freq[value] = 1
    else:
        height_freq[value] += 1

mode = 0
freq = 0
for key in height_freq:
    if mode == 0:
        # fill in here
    else:
        # fill in here
print height_freq
print mode
```