Assignment 4

Question 1 (Easy):

What will the following code output? You can use CoLab or shell to check out the output by running the code.

Using this as reference what would each of the following return:

x = {'key':'value', 'key1': 1, 'key2': sum([1,3,5]), 'key3': (1,2,5), 'key': 5.6}

- print(x)
- print(x['key6'])
- print(type(x['key2']))
- x['key3'][0] = 7
- print(x['key3'])
- for i, j in x.items():
  print(i)

Question 2 (Medium):

There are 7 problems in the following code which is trying to search for all occurrences of the target in a list of numbers. Find all 7 problems and create a good version that works. (Note: If you copy this code, the indentation might need to be fixed, but that is not one of the 7 problems)

```
def main():
  print("This is the main function")
  def circumference(diameter):
    return math.pi*diameter
  def area(diameter):
    return math.pi * (diameter//2)**2

  d == 5
  print("Circumference: ", circumferene(diameter))
  print("Area: ", area(diameter))
```

The correct output should look like this:

```
This is the main function

Circumference:  15.707963267948966
Area:  19.634954084936208
```

Question 3 (Hard):

Remember the backstory from basically all the assignments?
Let us say that you are a programmer for the Govt. of Delhi, India. The Govt. of Delhi has recently filed a law about even and odd number plates on cars in Delhi. Even and odd cars can stay on road alternatively to reduce pollution. They have already placed cameras to figure out the number on the plate of all cars. And you are given this number as an integer (n).

Question:

- Continuously take the input for car number(integer) and day ('odd' or 'even') from user again and again until the user enters -1 for car number
- For each car number and day entered by the user, store the combination of car number and day in one of the data structures we have learnt
- Use the previously created function from Assignment 3 and the stored values to check if all the stored car numbers are eligible to drive, and then print your answer
- An odd case scenario: if a car number is repeated in the inputs, then replace the existing day stored with the day from the latest car number entered

Example output:

```
Enter the car number: 1115
Enter 'even' for even day or 'odd' for odd day: even
Enter the car number: 1116
Enter 'even' for even day or 'odd' for odd day: even
Enter the car number: 1115
Enter 'even' for even day or 'odd' for odd day: odd
Enter the car number: 1118
Enter 'even' for even day or 'odd' for odd day: even
Enter the car number: -1
Hooray, you got an eligible number plate, you can drive the car with number 1115 today
Hooray, you got an eligible number plate, you can drive the car with number 1116 today
Hooray, you got an eligible number plate, you can drive the car with number 1118 today
```

HINT: You may use dictionary data type to solve this problem

Question 4 (Hard):

Given two number x and y find product (multiplication) using recursion.

HINT: multiplications is just adding n number m number of times i.e.

5 * 6 = (5 + 5 + 5 + 5 + 5 + 5) or (6 + 6 + 6 + 6 + 6)

Think of different cases before you start writing code!