Exercises

Answer the questions or complete the tasks outlined in bold below, use the specific method described if applicable.

ASSIGNMENT - 3

Name: Ranjanaa Y

Register Number: 412519104107

What is 7 to the power of 4?

```
In [1]: pow(7,4)

Out[1]: 2401
```

Split this string: s = "Hi there Sam!" into a list.

```
In [4]: s="Hi there sam!"
        s=s.split()
        print(s)

        ['Hi', 'there', 'sam!']
```

Given the variables:

planet = "Earth"

diameter = 12742

Use .format() to print the following string:

The diameter of Earth is 12742 kilometers.

```
In [3]: planet="Earth"
        diameter=12742
        print('The diameter of {} is {} kilometers.' .format(planet,diameter));

        The diameter of Earth is 12742 kilometers.
```

Given this nested list, use indexing to grab the word "hello"

lst=[1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]

```
In [5]: lst=[1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
        print(lst[3][1][2])

        ['hello']
```

Given this nest dictionary, grab the word "hello". Be prepared, this will be annoying/tricky

d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}

```
In [6]: d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
        print(d['k1'][3]['tricky'][3]['target'][3])

        hello
```

What is the main difference between a tuple and a list?

TUPLE :

    1. Tuple is immutable.

    2. Tuple iteration is faster.

    3. Tuple is useful for read-only operations like accessing elements.

    4. Tuples consumes less memory.

    5. Tuples have less in-built methods.

    6. Tuples operations are safe.

LIST :

    1. List is mutable.

    2. List iteration is slower and is time consuming

    3. List is useful for insertion and deletion operations.

    4. List consumes more memory.

    5. List provides many in-built methods.

    6. List operations are more error prone.

Create a function that grabs the email website domain from a string in the form:

user@domain.com

So for example, passing "user@domain.com" would return: domain.com

```
In [8]: def domainGet(email):
            print("your domain is:"+email.split('@')[-1])
        email=input("enter email: ")
        domainGet(email)

        enter email: user@domain.com
        your domain is:domain.com
```

Create a basic function that returns True if the word 'dog' is contained in the input string. Don't worry about edge cases like a punctuation being attached to the word dog, but do account for capitalization

```
In [9]: def findDog(st):
            if 'dog' in st.lower():
                print("True")
            else:
                print("False")
        st="Is theer a dog here?"
        findDog(st)

        True
```

Create a function that counts the number of times the word "dog" occurs in a

string. Again ignore edge cases.

```
In [10]: value='dog dog do do du du dog dog';
         def countdogs(value):
           count=0
           for word in value.lower().split():
             if word=='dog' or word=='dogs':
               count=count+1
           print(count)
         countdogs(value)

         4
```

Problem

You are driving a little too fast, and a police officer stops you. Write a function to return one of 3 possible results: "No ticket", "Small ticket", or "Big Ticket". If your speed is 60 or less, the result is "No Ticket". If speed is between 61 and 80 inclusive, the result is "Small Ticket". If speed is 81 or more, the result is "Big Ticket". Unless it is your birthday (encoded as a boolean value in the parameters of the function) -- on your birthday, your speed can be 5 higher in all cases.

```
In [11]: def caught_speeding(speed, is_birthday):

             if is_birthday:
                 speeding = speed - 5
             else:
                 speeding = speed

             if speeding > 80:
                 return 'Big Ticket'
             elif speeding > 60:
                 return 'Small Ticket'
             else:
                 return 'No Ticket'
```

```
In [12]: def caught_speeding(speed, is_birthday):

             if is_birthday:
                 speeding = speed - 5
             else:
                 speeding = speed

             if speeding > 80:
                 return 'Big Ticket'
             elif speeding > 60:
                 return 'Small Ticket'
             else:
                 return 'No Ticket'
         caught_speeding(90,True)

Out[12]: 'Big Ticket'
```

```
In [13]: def caught_speeding(speed, is_birthday):

             if is_birthday:
                 speeding = speed - 5
             else:
                 speeding = speed

             if speeding > 80:
                 return 'Big Ticket'
             elif speeding > 60:
                 return 'Small Ticket'
             else:
                 return 'No Ticket'
         caught_speeding(81,True)

Out[13]: 'Small Ticket'
```

Create an employee list with basic salary values(at least 5 values for 5 employees) and using a for loop retrieve each employee salary and calculate total salary expenditure.

```
In [14]: salary=[10000,15000,14000,16000,19000]
         sum=0
         for i in range(0,len(salary)):
             sum=sum+salary[i]
         print(sum)

         74000
```

Create two dictionaries in Python:

First one to contain fields as Empid, Empname, Basic Pay

Second dictionary to contain fields as DeptName, DeptId.

Combine both dictionaries.

```
In [15]: dict1= {"EmpId": 100 ,"Empname":"Sharwin","Basicpay": 45000}
         dict2={"DeptName":"IT","DeptId":5}
         def merge(dict1, dict2):
             return(dict1.update(dict2))
         merge(dict1,dict2)
         print(dict1)

         {'EmpId': 100, 'Empname': 'Sharwin', 'Basicpay': 45000, 'DeptName': 'IT', 'DeptId': 5}
```