ASSIGNMENT 3

TITLE: DA Assignment 3 python
TEAM ID: PNT2022TMID27387
NAME: YOGESH J

1. ** What is 7 to the power of 4? **

   Ans: 2401

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

Ans:
```
def Convert(string):
    li = list(string.split(" "))
    return li
str1 = "Hi there Sam"
print(Convert(str1))
```

```
['Hi', 'there', 'Sam']
```

3.** Given the variables:**

```
planet = "Earth"
```

diameter = 12742

** Use .format() to print the following string: **

The diameter of Earth is 12742 kilometres.

```
Ans: planet = "Earth"
       diameter = 12742
        print( 'The diameter of {} is {} kilometers.' .format(planet,diame
ter));
```

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

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

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

5. ** 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']}]}]}

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

6. ** What is the main difference between a tuple and a list? **

Ans:

TUPLE:

- the tuple has static characteristics.
- tuples cannot be modified.the tuple is faster than the list because of static in nature.
- tuples are denoted as parenthesis.

LIST:

- The list is dynamic.
- that lists can be modified
- Lists are denoted by the square brackets but tuples are denoted as parenthesis.

7. ** 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

Ans:

```
def domainGet(email):
 print("Your domain is: " + email.split('@')[-1])

email = input("Please enter your email: >")
domainGet(email)
```

8. ** 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. **

Ans:

```
def findDog(st):
    if 'dog' in st.lower():
        print("True")
    else:
        print("False")
```

```
st = "Is there a dog here?"
findDog(st)
```

```
True
```

```
findDog('Is there a dog here?')
```

```
True
```

9. ** Create a function that counts the number of times the word "dog" occurs in a string. Again ignore edge cases. **

Ans:

```
   value = 'This dog runs faster than the other dog dude!';

def countdogs(value):
    count = 0
    for word in value.lower().split():
        if word == 'dog' or word == 'dogs':
            count = count + 1
            print(count)

countdogs(value)
```

10. ### 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. **

```
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'

Ans:

```
def caught_speeding(speed, is_birthday):
    pass

caught_speeding(81,True)

linkcode
caught_speeding(81,False)
```

11. Create an employee list with basic salary values(at least 5 values for 5 employees) and using a for loop retreive each employee salary and calculate total salary expenditure. Create two dictionaries in Python:

First one to contain fields as Empid, Empname, Basicpay,Second dictionary to contain fields as DeptName, DeptId.Combine both dictionaries.

```
import pandas as pd

import numpy as np
```

```
employee = pd.read_csv("Employees.csv")

employee['BIRTHDAY']=pd.to_datetime(employee['BIRTHDAY'])
#Calculate an array of calculated column values, group records by them, and calculate the average salary
years_salary = employee.groupby(np.floor((employee['BIRTHDAY'].dt.year-1900)/10)).SALARY.mean()

print(years_salary)
```