LAB ASSIGNMENT – 2

NAME : PRATHAPANI SATWIKA

REG.NO. : 20BCD7160

Q1.Given a string, return the sum and average of the digits that appear in the string, ignoring all other characters Given:

str1 = "English = 78 Science = 83 Math = 68 History = 65"

Expected Outcome:

sum is 294 average is 73.5

CODE :

```
str1 = "English = 78 Science = 83 Math = 68 History = 65"
str2 = str1.split(' ')
nums = 0
count= 0
for i in str2:
    if i.isdigit():
        nums+= int(i)
        count+=1


print('sum is',nums)
print('average is',nums/count)
```

OUTPUT :

```
>>>
=== RESTART: C:/Users/Dileep/Documents/3-1/Data warehousing lab/lab2.1.py ===
sum is 294
average is 73.5
>>>
```

Q2.Given a two list. Create a third list by picking an odd-index element from the first list and even index elements from second.

For Example:

listOne = [3, 6, 9, 12, 15, 18, 21] listTwo = [4, 8, 12, 16, 20, 24, 28]

Expected Output:

Element at odd-index positions from list one [6, 12, 18]

Element at even-index positions from list two [4, 12, 20, 28]

Printing Final third list [6, 12, 18, 4, 12, 20, 28]

CODE :

```
listOne = [3, 6, 9, 12, 15, 18, 21]
listTwo = [4, 8, 12, 16, 20, 24, 28]
listThree = list()

oddElements = listOne[1::2]
print("Element at odd-index positions from list one")
print(oddElements)

EvenElement = listTwo[0::2]
print("Element at even-index positions from list two")
print(EvenElement)

print("Final third list")
listThree.extend(oddElements)
listThree.extend(EvenElement)
print(listThree)
```

OUTPUT :

```
>>>
=== RESTART: C:/Users/Dileep/Documents/3-1/Data warehousing lab/lab2.2.py ===
Element at odd-index positions from list one
[6, 12, 18]
Element at even-index positions from list two
[4, 12, 20, 28]
Final third list
[6, 12, 18, 4, 12, 20, 28]
>>>
```

Q3.Given a two list of equal size create a set such that it shows the element from both lists in the pair Expected Output:

First List [2, 3, 4, 5, 6, 7, 8]

Second List [4, 9, 16, 25, 36, 49, 64]

Result is {(6, 36), (8, 64), (4, 16), (5, 25), (3, 9), (7, 49), (2, 4)}

CODE :

```
first_list = [2, 3, 4, 5, 6, 7, 8]
print("First List ", first_list)


second_list = [4, 9, 16, 25, 36, 49, 64]
print("Second List ", second_list)


result = zip(first_list, second_list)
result_set = set(result)
print('Result is',result_set)
```

```
=== RESTART: C:/Users/Dileep/Documents/3-1/Data warehousing lab/lab2.3.py ===
First List  [2, 3, 4, 5, 6, 7, 8]
Second List  [4, 9, 16, 25, 36, 49, 64]
Result is {(6, 36), (8, 64), (4, 16), (5, 25), (3, 9), (7, 49), (2, 4)}
>>>
```

Q4.Given a dictionary get all values from the dictionary and add it in a list but don't add duplicates.

speed ={'jan':47, 'feb':52, 'march':47, 'April':44, 'May':52, 'June':53, 'july':54, 'Aug':44, 'Sept':54} Expected Outcome: [47, 52, 44, 53, 54]

CODE :

```
speed = {'jan': 47, 'feb': 52, 'march': 47, 'April': 44, 'May': 52, 'June': 53,
        'july': 54, 'Aug': 44, 'Sept': 54}


print("Dictionary's values - ", speed.values())


speed_list = list()


# iterate dict values
for val in speed.values():
    # check if value not present in a list
    if val not in speed_list:
        speed_list.append(val)
print("unique list", speed_list)
```

OUTPUT :

```
=== RESTART: C:/Users/Dileep/Documents/3-1/Data warehousing lab/lab2.4.py ===
Dictionary's values -  dict_values([47, 52, 47, 44, 52, 53, 54, 44, 54])
unique list [47, 52, 44, 53, 54]
>>>
```

Q5. Remove duplicate from a list and create a tuple and find the minimum and maximum number. For Example:

sampleList = [87, 45, 41, 65, 94, 41, 99, 94] Expected Outcome:

unique items [87, 45, 41, 65, 99] tuple (87, 45, 41, 65, 99) min: 41 max: 99

CODE :

```
sample_list = [87, 45, 41, 65, 94, 41, 99, 94]


print("Original list", sample_list)


sample_list = list(set(sample_list))
print("unique list", sample_list)


t = tuple(sample_list)
print("tuple ", t)


print("Minimum number is: ", min(t))
print("Maximum number is: ", max(t))
```

OUTPUT :

```
=== RESTART: C:/Users/Dileep/Documents/3-1/Data warehousing lab/lab2.5.py ===
Original list [87, 45, 41, 65, 94, 41, 99, 94]
unique list [65, 99, 41, 45, 87, 94]
tuple  (65, 99, 41, 45, 87, 94)
Minimum number is:  41
Maximum number is:  99
>>>
```