MACHINE LEARNING
ASSIGNMENT1

NAME:ANJALI ERRA(700740323)
Github link: https://github.com/Anjali555-erra/MLAssignment1.git
Video Link: https://drive.google.com/file/d/134xajI-XZX_nmYqdNUoiiSlPGKnenKW1/view?usp=share_link

Question 1

The following is a list of 10 students ages:
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

- Sort the list and find the min and max age
- Add the min age and the max age again to the list
- Find the median age (one middle item or two middle items divided by two)
- Find the average age (sum of all items divided by their number)
- Find the range of the ages (max minus min)

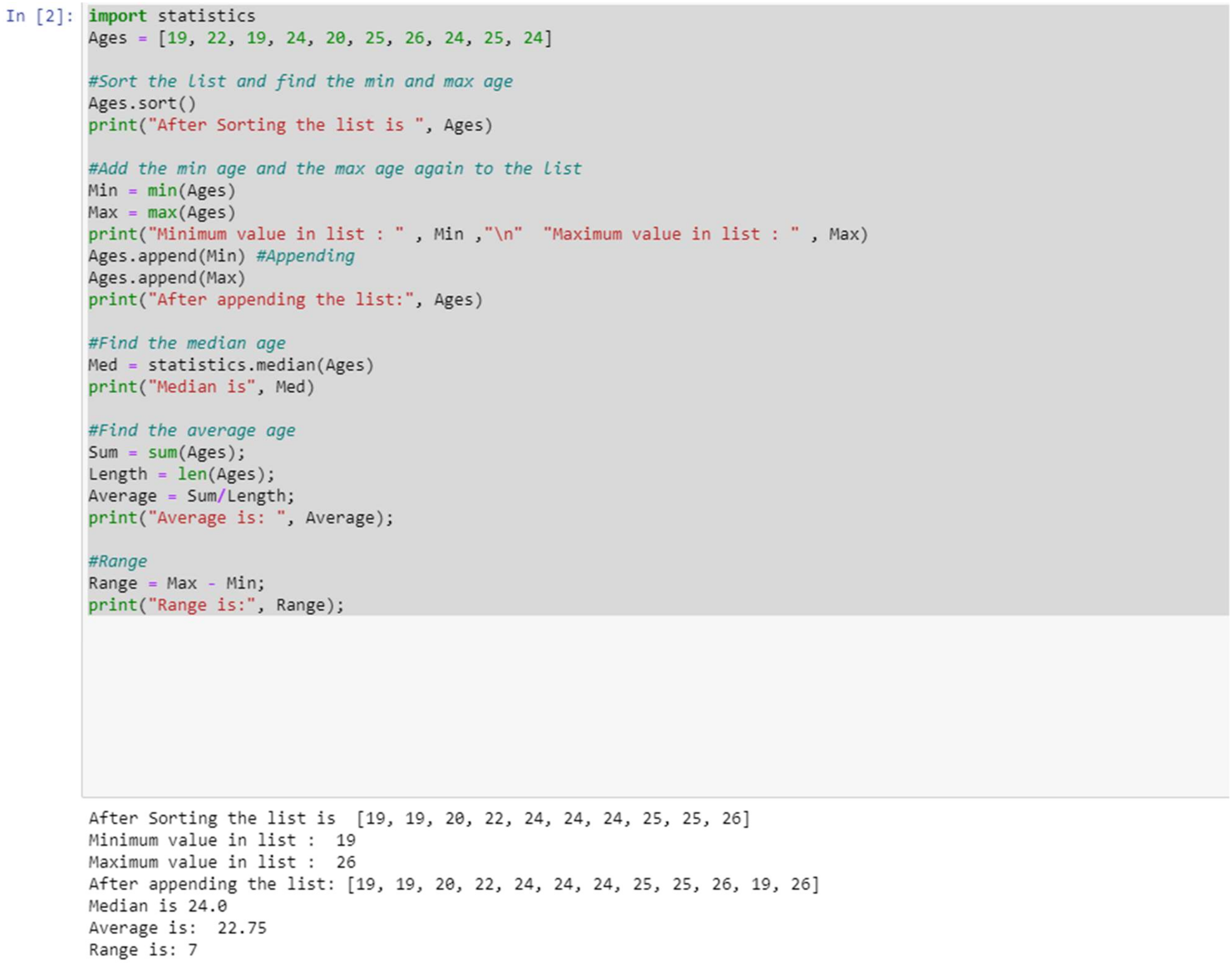
Source Code:

```
import statistics
Ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

#Sort the list and find the min and max ages
Ages.sort()
print("After Sorting the list is ", Ages)

#Add the min age and the max age again to the list
Min = min(Ages)
Max = max(Ages)
print("Minimum value in list : " , Min ,"\n"  "Maximum value in list : " , Max)
Ages.append(Min) #Appending
Ages.append(Max)
print("After appending the list:", Ages)

#Find the median age
Med = statistics.median(Ages)
print("Median is", Med)

#Find the average age
Sum = sum(Ages);
Length = len(Ages);
Average = Sum/Length;
print("Average is: ", Average);
```

```
#Range
Range = Max - Min;
print("Range is:", Range);
```

Output:

```
After Sorting the list is  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Minimum value in list :  19
Maximum value in list :  26
After appending the list: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Median is 24.0
Average is:  22.75
Range is: 7
```



```
In [2]: import statistics
        Ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

        #Sort the List and find the min and max age
        Ages.sort()
        print("After Sorting the list is ", Ages)

        #Add the min age and the max age again to the List
        Min = min(Ages)
        Max = max(Ages)
        print("Minimum value in list : " , Min ,"\n"  "Maximum value in list : " , Max)
        Ages.append(Min) #Appending
        Ages.append(Max)
        print("After appending the list:", Ages)

        #Find the median age
        Med = statistics.median(Ages)
        print("Median is", Med)

        #Find the average age
        Sum = sum(Ages);
        Length = len(Ages);
        Average = Sum/Length;
        print("Average is: ", Average);

        #Range
        Range = Max - Min;
        print("Range is:", Range);
```

```
After Sorting the list is  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Minimum value in list :  19
Maximum value in list :  26
After appending the list: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Median is 24.0
Average is:  22.75
Range is: 7
```

Question 2:

- Create an empty dictionary called dog
- Add name, color, breed, legs, age to the dog dictionary
- Create a student dictionary and add first_name, last_name, gender, age, marital status, skills, country, city and address as keys for the dictionary

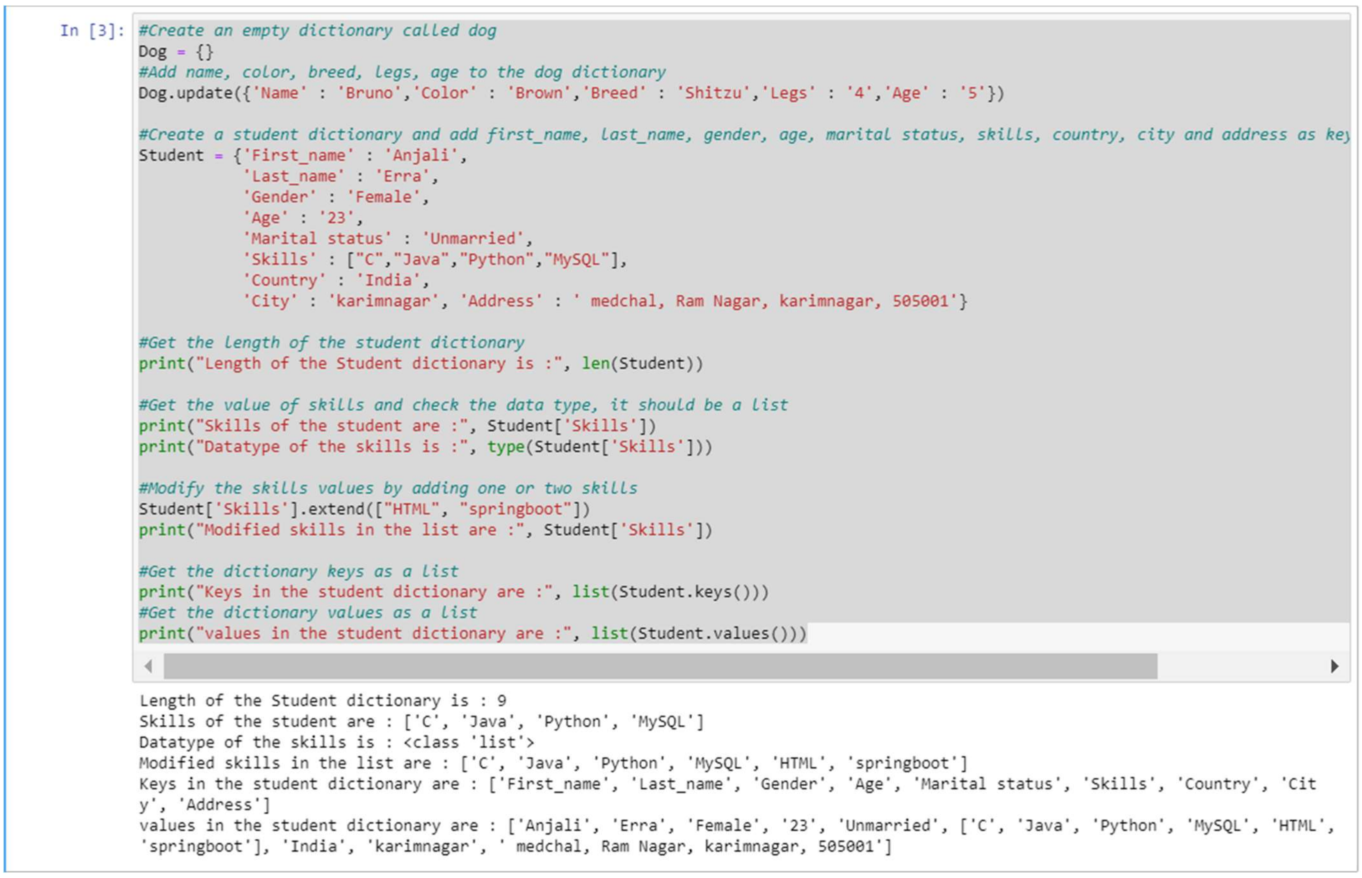
- Get the length of the student dictionary
- Get the value of skills and check the data type, it should be a list
- Modify the skills values by adding one or two skills
- Get the dictionary keys as a list
- Get the dictionary values as a list

Source code:

```
#Create an empty dictionary called dog
Dog = {}
#Add name, color, breed, legs, age to the dog dictionary
Dog.update({'Name' : 'Bruno','Color' : 'Brown','Breed' : 'Shitzu','Legs' : '4','Age' : '5'})

#Create a student dictionary and add first_name, last_name, gender, age, marital status, skills, country, city and
address as keys for the dictionary
Student = {'First_name' : 'Anjali',
          'Last_name' : 'Erra',
          'Gender' : 'Female',
          'Age' : '23',
          'Marital status' : 'Unmarried',
          'Skills' : ["C","Java","Python","MySQL"],
          'Country' : 'India',
          'City' : 'karimnagar', 'Address' : ' medchal, Ram Nagar, karimnagar, 505001'}

#Get the length of the student dictionary
print("Length of the Student dictionary is :", len(Student))

#Get the value of skills and check the data type, it should be a list
print("Skills of the student are :", Student['Skills'])
print("Datatype of the skills is :", type(Student['Skills']))

#Modify the skills values by adding one or two skills
Student['Skills'].extend(["HTML", "springboot"])
print("Modified skills in the list are :", Student['Skills'])

#Get the dictionary keys as a list
print("Keys in the student dictionary are :", list(Student.keys()))
#Get the dictionary values as a list
print("values in the student dictionary are :", list(Student.values()))
```

Output:

```
Length of the Student dictionary is : 9
Skills of the student are : ['C', 'Java', 'Python', 'MySQL']
Datatype of the skills is : <class 'list'>
Modified skills in the list are : ['C', 'Java', 'Python', 'MySQL', 'HTML', 'springboot']
Keys in the student dictionary are : ['First_name', 'Last_name', 'Gender', 'Age', 'Marital status', 'Skills', 'Country', 'Cit
y', 'Address']
values in the student dictionary are : ['Anjali', 'Erra', 'Female', '23', 'Unmarried', ['C', 'Java', 'Python', 'MySQL', 'HTM
L', 'springboot'], 'India', 'karimnagar', ' medchal, Ram Nagar, karimnagar, 505001']
```



In [3]:

```
#Create an empty dictionary called dog
Dog = {}
#Add name, color, breed, legs, age to the dog dictionary
Dog.update({'Name' : 'Bruno','Color' : 'Brown','Breed' : 'Shitzu','Legs' : '4','Age' : '5'})

#Create a student dictionary and add first_name, last_name, gender, age, marital status, skills, country, city and address as key
Student = {'First_name' : 'Anjali',
           'Last_name' : 'Erra',
           'Gender' : 'Female',
           'Age' : '23',
           'Marital status' : 'Unmarried',
           'Skills' : ["C","Java","Python","MySQL"],
           'Country' : 'India',
           'City' : 'karimnagar', 'Address' : ' medchal, Ram Nagar, karimnagar, 505001'}

#Get the length of the student dictionary
print("Length of the Student dictionary is :", len(Student))

#Get the value of skills and check the data type, it should be a list
print("Skills of the student are :", Student['Skills'])
print("Datatype of the skills is :", type(Student['Skills']))

#Modify the skills values by adding one or two skills
Student['Skills'].extend(["HTML", "springboot"])
print("Modified skills in the list are :", Student['Skills'])

#Get the dictionary keys as a list
print("Keys in the student dictionary are :", list(Student.keys()))
#Get the dictionary values as a list
print("values in the student dictionary are :", list(Student.values()))
```

```
Length of the Student dictionary is : 9
Skills of the student are : ['C', 'Java', 'Python', 'MySQL']
Datatype of the skills is : <class 'list'>
Modified skills in the list are : ['C', 'Java', 'Python', 'MySQL', 'HTML', 'springboot']
Keys in the student dictionary are : ['First_name', 'Last_name', 'Gender', 'Age', 'Marital status', 'Skills', 'Country', 'Cit
y', 'Address']
values in the student dictionary are : ['Anjali', 'Erra', 'Female', '23', 'Unmarried', ['C', 'Java', 'Python', 'MySQL', 'HTML',
'springboot'], 'India', 'karimnagar', ' medchal, Ram Nagar, karimnagar, 505001']
```

Question 3:

- Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)
- Join brothers and sisters tuples and assign it to siblings
- How many siblings do you have?

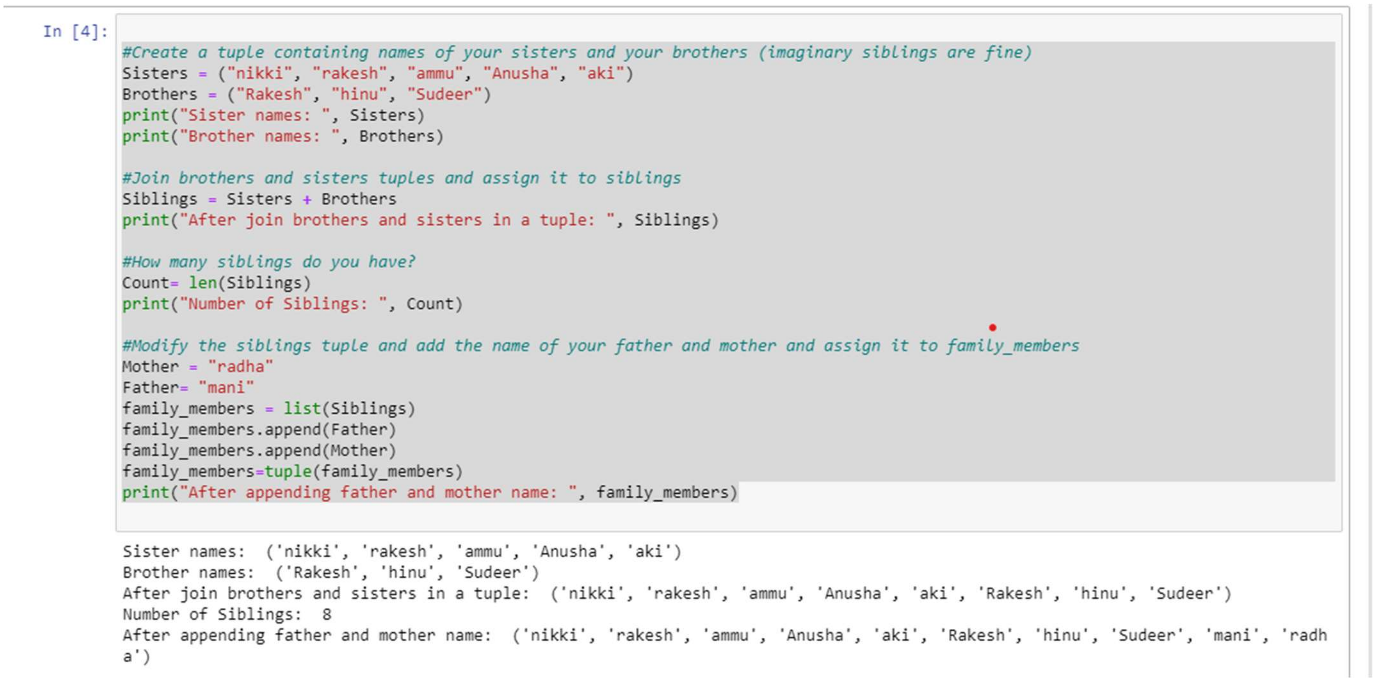
- Modify the siblings tuple and add the name of your father and mother and assign it to family_members

Source code:

```
#Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)
Sisters = ("nikki", "rakesh", "ammu", "Anusha", "aki")
Brothers = ("Rakesh", "hinu", "Sudeer")
print("Sister names: ", Sisters)
print("Brother names: ", Brothers)

#Join brothers and sisters tuples and assign it to siblings
Siblings = Sisters + Brothers
print("After join brothers and sisters in a tuple: ", Siblings)

#How many siblings do you have?
Count= len(Siblings)
print("Number of Siblings: ", Count)

#Modify the siblings tuple and add the name of your father and mother and assign it to family_members
Mother = "radha"
Father= "mani"
family_members = list(Siblings)
family_members.append(Father)
family_members.append(Mother)
family_members=tuple(family_members)
print("After appending father and mother name: ", family_members)
```

Output:

```
Sister names:  ('nikki', 'rakesh', 'ammu', 'Anusha', 'aki')
Brother names:  ('Rakesh', 'hinu', 'Sudeer')
After join brothers and sisters in a tuple:  ('nikki', 'rakesh', 'ammu', 'Anusha', 'aki', 'Rakesh', 'hinu', 'Sudeer')
Number of Siblings:  8
After appending father and mother name:  ('nikki', 'rakesh', 'ammu', 'Anusha', 'aki', 'Rakesh', 'hinu', 'Sudeer', 'mani', 'r
adha')
```



```
In [4]:
#Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)
Sisters = ("nikki", "rakesh", "ammu", "Anusha", "aki")
Brothers = ("Rakesh", "hinu", "Sudeer")
print("Sister names: ", Sisters)
print("Brother names: ", Brothers)

#Join brothers and sisters tuples and assign it to siblings
Siblings = Sisters + Brothers
print("After join brothers and sisters in a tuple: ", Siblings)

#How many siblings do you have?
Count= len(Siblings)
print("Number of Siblings: ", Count)

#Modify the siblings tuple and add the name of your father and mother and assign it to family_members
Mother = "radha"
Father= "mani"
family_members = list(Siblings)
family_members.append(Father)
family_members.append(Mother)
family_members=tuple(family_members)
print("After appending father and mother name: ", family_members)
```

```
Sister names:  ('nikki', 'rakesh', 'ammu', 'Anusha', 'aki')
Brother names:  ('Rakesh', 'hinu', 'Sudeer')
After join brothers and sisters in a tuple:  ('nikki', 'rakesh', 'ammu', 'Anusha', 'aki', 'Rakesh', 'hinu', 'Sudeer')
Number of Siblings:  8
After appending father and mother name:  ('nikki', 'rakesh', 'ammu', 'Anusha', 'aki', 'Rakesh', 'hinu', 'Sudeer', 'mani', 'radh
a')
```

Question 4:

it_companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'} A = {19, 22, 24, 20, 25, 26}
B = {19, 22, 20, 25, 26, 24, 28, 27}
age = [22, 19, 24, 25, 26, 24, 25, 24]

- • Find the length of the set it_companies
- • Add 'Twitter' to it_companies
- • Insert multiple IT companies at once to the set it_companies
- • Remove one of the companies from the set it_companies
- • What is the difference between remove and discard
- • Join A and B
- • Find A intersection B
- • Is A subset of B
- • Are A and B disjoint sets
- • Join A with B and B with A
- • What is the symmetric difference between A and B
- • Delete the sets completely
- • Convert the ages to a set and compare the length of the list and the set.

Source code:

IT_Companies = {'Facebook', 'Google', 'Microsoft', 'Apple', 'IBM', 'Oracle', 'Amazon'}

A = {19, 22, 24, 20, 25, 26}

B = {19, 22, 20, 25, 26, 24, 28, 27}

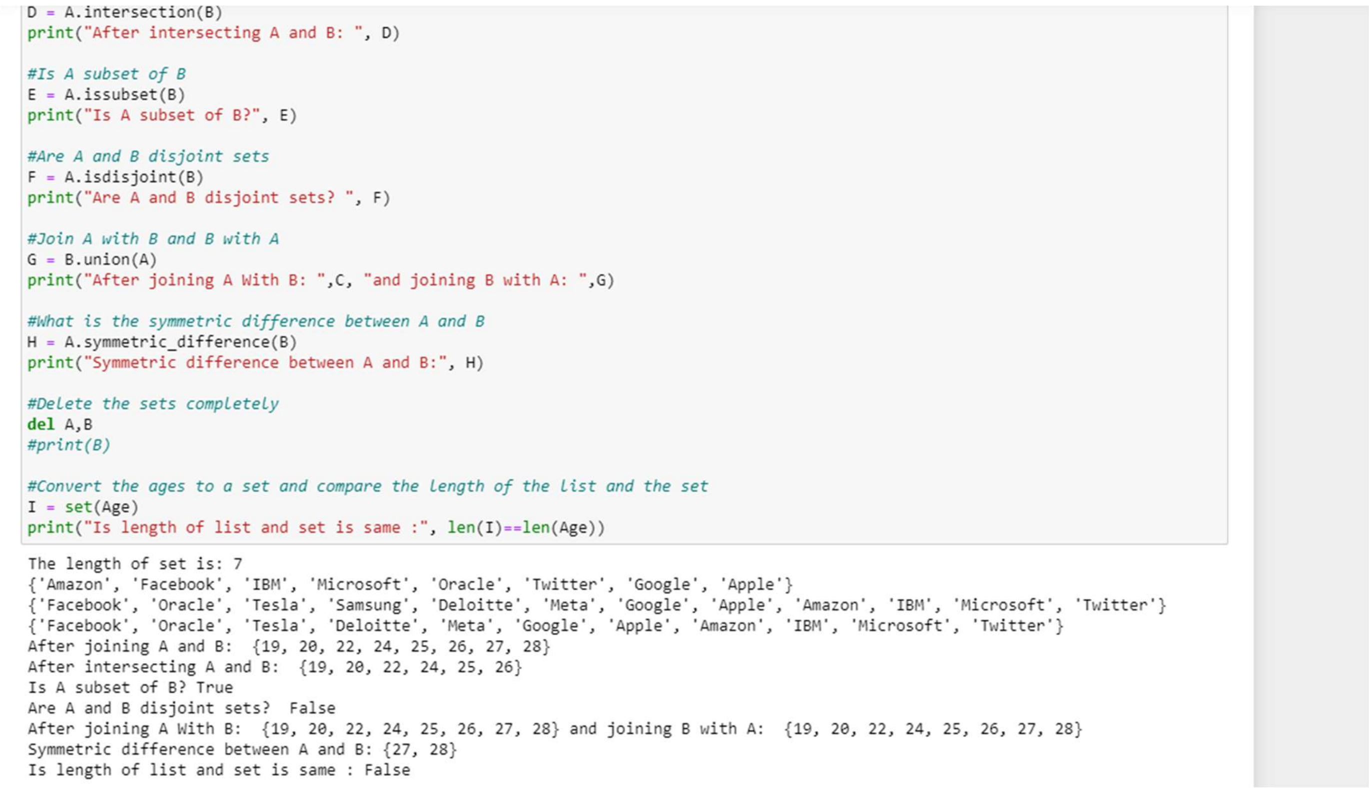
Age = [22, 19, 24, 25, 26, 24, 25, 24]

```
#Find the length of the set it_companies
print("The length of set is:", len(IT_Companies))

#Add 'Twitter' to it_companies
IT_Companies.add("Twitter")
print(IT_Companies)

#Insert multiple IT companies at once to the set it_companies
Multiple_ITcompanies= ["Tesla","Samsung", "Deloitte", "Meta"]
IT_Companies.update(Multiple_ITcompanies)
print(IT_Companies)

#Remove one of the companies from the set it_companies
IT_Companies.remove("Samsung")
print(IT_Companies)

#What is the difference between remove and discard
#Remove: If the item to remove does not exist, remove() will raise an error
#Discard: If the item to remove does not exist, discard() will NOT raise an error

#Join A and B
C = A.union(B)
print("After joining A and B: ", C)

#Find A intersection B
D = A.intersection(B)
print("After intersecting A and B: ", D)

#Is A subset of B
```

```
E = A.issubset(B)

print("Is A subset of B?", E)


#Are A and B disjoint sets

F = A.isdisjoint(B)

print("Are A and B disjoint sets? ", F)


#Join A with B and B with A

G = B.union(A)

print("After joining A With B: ",C, "and joining B with A: ",G)


#What is the symmetric difference between A and B

H = A.symmetric_difference(B)

print("Symmetric difference between A and B:", H)


#Delete the sets completely

del A,B

#print(B)


#Convert the ages to a set and compare the length of the list and the set

I = set(Age)

print("Is length of list and set is same :", len(I)==len(Age))
```

Output:

```
The length of set is: 7
{'Amazon', 'Facebook', 'IBM', 'Microsoft', 'Oracle', 'Twitter', 'Google', 'Apple'}
{'Facebook', 'Oracle', 'Tesla', 'Samsung', 'Deloitte', 'Meta', 'Google', 'Apple', 'Amazon', 'IBM', 'Microsoft', 'Twitter'}
{'Facebook', 'Oracle', 'Tesla', 'Deloitte', 'Meta', 'Google', 'Apple', 'Amazon', 'IBM', 'Microsoft', 'Twitter'}
After joining A and B:  {19, 20, 22, 24, 25, 26, 27, 28}
After intersecting A and B:  {19, 20, 22, 24, 25, 26}
Is A subset of B? True
Are A and B disjoint sets?  False
After joining A With B:  {19, 20, 22, 24, 25, 26, 27, 28} and joining B with A:  {19, 20, 22, 24, 25, 26, 27, 28}
Symmetric difference between A and B: {27, 28}
Is length of list and set is same : False
```



```
D = A.intersection(B)
print("After intersecting A and B: ", D)

#Is A subset of B
E = A.issubset(B)
print("Is A subset of B?", E)

#Are A and B disjoint sets
F = A.isdisjoint(B)
print("Are A and B disjoint sets? ", F)

#Join A with B and B with A
G = B.union(A)
print("After joining A With B: ",C, "and joining B with A: ",G)

#What is the symmetric difference between A and B
H = A.symmetric_difference(B)
print("Symmetric difference between A and B:", H)

#Delete the sets completely
del A,B
#print(B)

#Convert the ages to a set and compare the length of the list and the set
I = set(Age)
print("Is length of list and set is same :", len(I)==len(Age))
```

```
The length of set is: 7
{'Amazon', 'Facebook', 'IBM', 'Microsoft', 'Oracle', 'Twitter', 'Google', 'Apple'}
{'Facebook', 'Oracle', 'Tesla', 'Samsung', 'Deloitte', 'Meta', 'Google', 'Apple', 'Amazon', 'IBM', 'Microsoft', 'Twitter'}
{'Facebook', 'Oracle', 'Tesla', 'Deloitte', 'Meta', 'Google', 'Apple', 'Amazon', 'IBM', 'Microsoft', 'Twitter'}
After joining A and B:  {19, 20, 22, 24, 25, 26, 27, 28}
After intersecting A and B:  {19, 20, 22, 24, 25, 26}
Is A subset of B? True
Are A and B disjoint sets?  False
After joining A With B:  {19, 20, 22, 24, 25, 26, 27, 28} and joining B with A:  {19, 20, 22, 24, 25, 26, 27, 28}
Symmetric difference between A and B: {27, 28}
Is length of list and set is same : False
```

Question 5:

The radius of a circle is 30 meters.

- Calculate the area of a circle and assign the value to a variable name of _area_of_circle_
- Calculate the circumference of a circle and assign the value to a variable name of _circum_of_circle_
- Take radius as user input and calculate the area.

Source code:

radius = 30

pi = 3.14

#Calculate the area of a circle

_area_of_circle_= pi*radius*radius

print("Area of the circle is: ", _area_of_circle_)

#Calculate the circumference of a circle

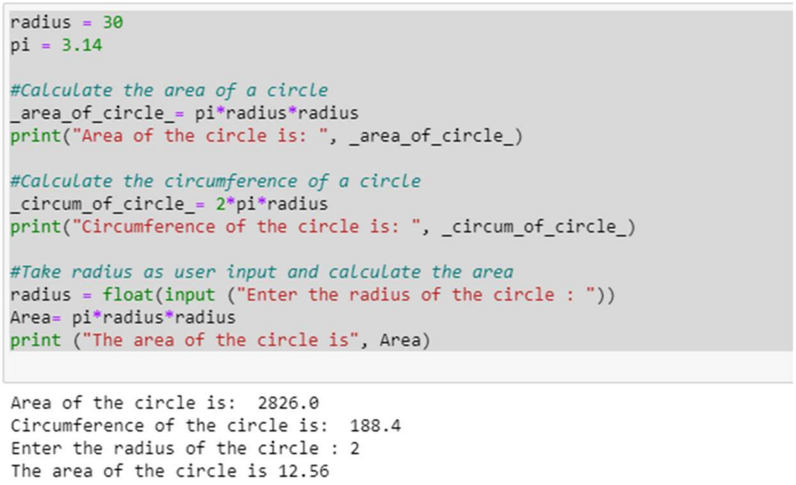
_circum_of_circle_= 2*pi*radius

print("Circumference of the circle is: ", _circum_of_circle_)

```
#Take radius as user input and calculate the area

radius = float(input ("Enter the radius of the circle : "))

Area= pi*radius*radius

print ("The area of the circle is", Area)
```

output:

```
Area of the circle is:  2826.0
Circumference of the circle is:  188.4
Enter the radius of the circle : 2
The area of the circle is 12.56
```



```
radius = 30
pi = 3.14

#Calculate the area of a circle
_area_of_circle_= pi*radius*radius
print("Area of the circle is: ", _area_of_circle_)

#Calculate the circumference of a circle
_circum_of_circle_= 2*pi*radius
print("Circumference of the circle is: ", _circum_of_circle_)

#Take radius as user input and calculate the area
radius = float(input ("Enter the radius of the circle : "))
Area= pi*radius*radius
print ("The area of the circle is", Area)
```

```
Area of the circle is:  2826.0
Circumference of the circle is:  188.4
Enter the radius of the circle : 2
The area of the circle is 12.56
```

Question 6:

"I am a teacher and I love to inspire and teach people"

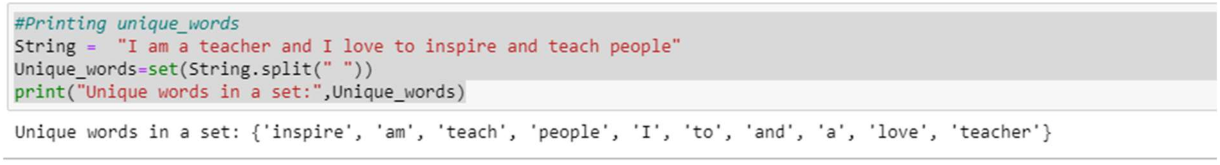
- How many unique words have been used in the sentence? Use the split methods and set to get the unique words.

Source code:

```
#Printing unique_words

String =  "I am a teacher and I love to inspire and teach people"

Unique_words=set(String.split(" "))

print("Unique words in a set:",Unique_words)
```

output:

```
Unique words in a set: {'inspire', 'am', 'teach', 'people', 'I', 'to', 'and', 'a', 'love', 'teacher'}
```



```
#Printing unique_words
String =  "I am a teacher and I love to inspire and teach people"
Unique_words=set(String.split(" "))
print("Unique words in a set:",Unique_words)
```

```
Unique words in a set: {'inspire', 'am', 'teach', 'people', 'I', 'to', 'and', 'a', 'love', 'teacher'}
```

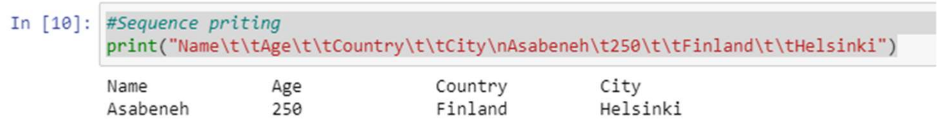
Question 7:

Use a tab escape sequence to get the following lines.
Name Age Country City Asabeneh 250 Finland Helsinki

Source code:
#Sequence priting
print("Name\t\tAge\t\tCountry\t\tCity\nAsabeneh\t250\t\tFinland\t\tHelsinki")

output:

```
Name                Age                 Country             City
Asabeneh 250                Finland             Helsinki
```



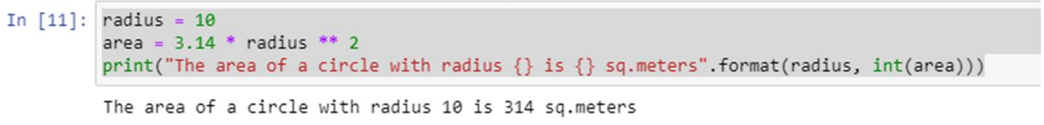
```
In [10]: #Sequence priting
         print("Name\t\tAge\t\tCountry\t\tCity\nAsabeneh\t250\t\tFinland\t\tHelsinki")
```

```
Name            Age             Country         City
Asabeneh        250             Finland         Helsinki
```

Question 8:
Use the string formatting method to display the following:
radius = 10
area = 3.14 * radius ** 2
“The area of a circle with radius 10 is 314 meters square.”

Source code:
radius = 10
area = 3.14 * radius ** 2
print("The area of a circle with radius {} is {} sq.meters".format(radius, int(area)))

output:
The area of a circle with radius 10 is 314 sq.meters



```
In [11]: radius = 10
         area = 3.14 * radius ** 2
         print("The area of a circle with radius {} is {} sq.meters".format(radius, int(area)))
```

```
The area of a circle with radius 10 is 314 sq.meters
```

Question 9:

Write a program, which reads weights (lbs.) of N students into a list and convert these weights to kilograms in a separate list using Loop. N: No of students (Read input from user)
Ex: L1: [150, 155, 145, 148]
Output: [68.03, 70.3, 65.77, 67.13]

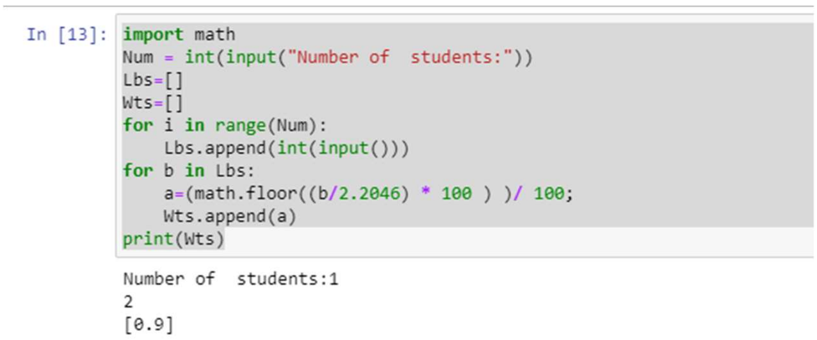
Source code:

import math

```
Num = int(input("Number of  students:"))
Lbs=[]
Wts=[]
for i in range(Num):
    Lbs.append(int(input()))
for b in Lbs:
    a=(math.floor((b/2.2046) * 100 ) )/ 100;
    Wts.append(a)
print(Wts)
```

output:

```
Number of  students:1
2
[0.9]
```



```
In [13]: import math
         Num = int(input("Number of  students:"))
         Lbs=[]
         Wts=[]
         for i in range(Num):
             Lbs.append(int(input()))
         for b in Lbs:
             a=(math.floor((b/2.2046) * 100 ) )/ 100;
             Wts.append(a)
         print(Wts)

         Number of  students:1
         2
         [0.9]
```

Question 10:

The diagram below shows a dataset with 2 classes and 8 data points, each with only one feature value, labeled f. Note that there are two data points with the same feature value of 6. These are shown as two x's one above the other. Provide stepwise mathematical solution, do not write code for it.

1. Divide this data equally into two parts. Use first part as training and second part as testing. Using KNN classifier, for K=3, what would be the predicted outputs for the test samples? Show how you arrived at your answer.

2. Compute the confusion matrix for this and calculate accuracy, sensitivity and specificity values.

Teacher's Sign/Remarks

10) Given dataset ie

F	label
1	0
2	0
3	1
6	1
6	1
7	0
10	0
11	0

Divide the above dataset equally and then taken one part as train data and Second one as test data

10 a) we have to find the predicted olps for test set with the above train set using knn for K=3

* In knn we will be using Euclidian distance between data points

$$D = \sqrt{(a_2 - a_1)^2}$$

Train

F	label
1	0
2	0
3	1
6	1

F	label	Predicted label	0	3	4	5
6	1					
7	0					
10	0					
11	0					

$d = \sqrt{(x_2 - x_1)^2}$

for data points 6 and train point

$d = \sqrt{(6-6)^2} = 0$

for data point 6 & train point 3

$d = \sqrt{(6-3)^2} = 3$

for data point 6 and train point 2

$d = \sqrt{(6-2)^2} = 4$

for data point 6 & train point 1

$d = \sqrt{(6-1)^2} = 5$

According to knn we have to take three nearest values ie) (1, 1, 0)

* So the predicted label for test point 6 is "1"

* In the same way predicted values for test points 7, 10, 11 is 1, 1, 1

* Below is the test data set for predicted label

F	label	Predicted label
6	1	1
7	0	1
10	0	1
11	0	1

10) b) True positive (Tp):- when actual value and predicted value is positive (1) Then it is called 'Tp'

confusion matrix

F	label	predicted label	$T_p \mid T_N \mid F_p \mid F_N$
6	1	1	Tp
7	0	1	Fp
10	0	1	Fp
11	0	1	Fp

value of $T_p = 1$, value of $T_N = 0$

value of $F_p = 3$, value of $F_N = 0$

Accuracy :- $\dfrac{T_p + T_N}{T_p + T_N + F_p + F_N} = \dfrac{1+0}{1+0+3+0} = 0.25$

Sensitivity : $\dfrac{T_p}{T_p + F_N} = \dfrac{1}{1+0} = 1$

Specificity = $\dfrac{T_N}{F_p + T_N} = \dfrac{0}{3+0} = 0$