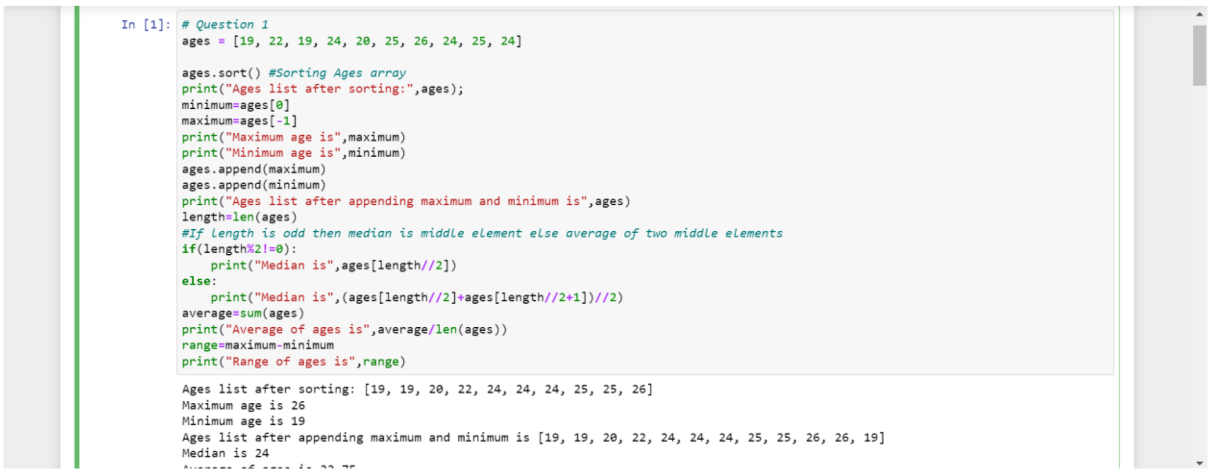
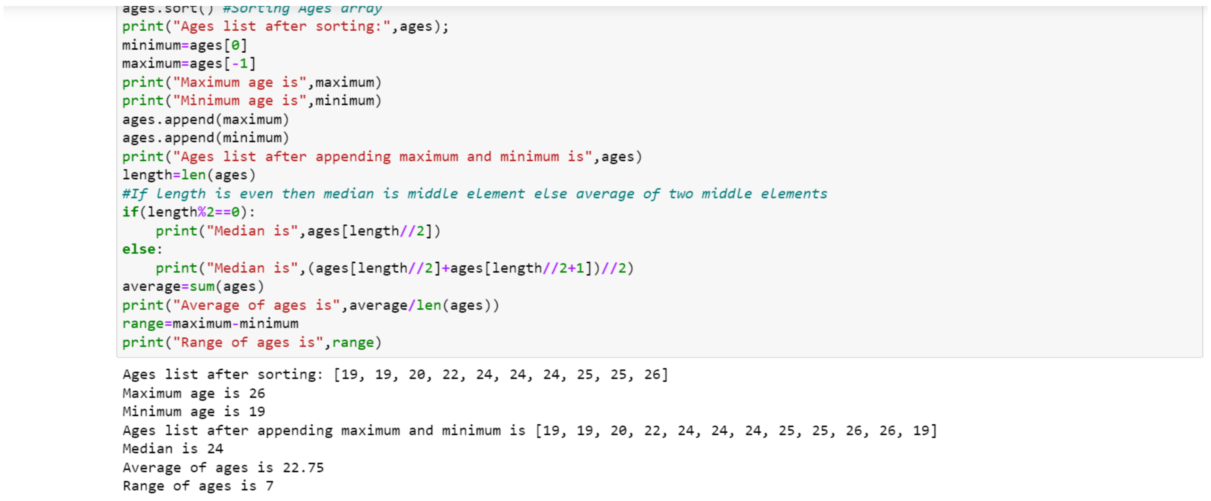
ASSIGNMENT-1



```
In [1]: # Question 1
        ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

        ages.sort() #Sorting Ages array
        print("Ages list after sorting:",ages);
        minimum=ages[0]
        maximum=ages[-1]
        print("Maximum age is",maximum)
        print("Minimum age is",minimum)
        ages.append(maximum)
        ages.append(minimum)
        print("Ages list after appending maximum and minimum is",ages)
        length=len(ages)
        #If length is odd then median is middle element else average of two middle elements
        if(length%2!=0):
            print("Median is",ages[length//2])
        else:
            print("Median is",(ages[length//2]+ages[length//2+1])//2)
        average=sum(ages)
        print("Average of ages is",average/len(ages))
        range=maximum-minimum
        print("Range of ages is",range)

        Ages list after sorting: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
        Maximum age is 26
        Minimum age is 19
        Ages list after appending maximum and minimum is [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 26, 19]
        Median is 24
```

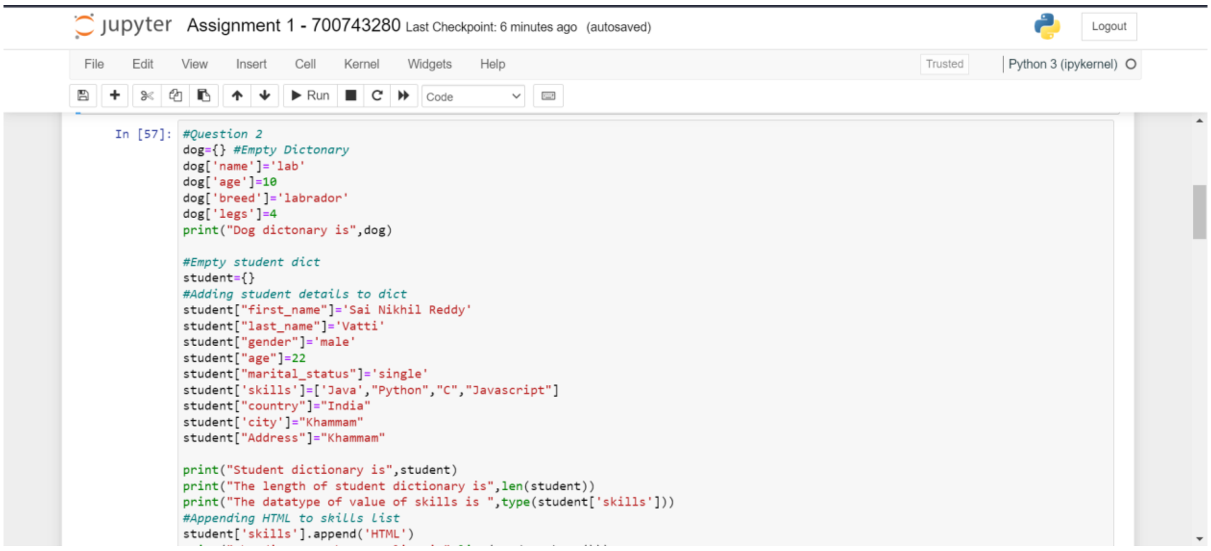


```
ages.sort() #Sorting Ages array
print("Ages list after sorting:",ages);
minimum=ages[0]
maximum=ages[-1]
print("Maximum age is",maximum)
print("Minimum age is",minimum)
ages.append(maximum)
ages.append(minimum)
print("Ages list after appending maximum and minimum is",ages)
length=len(ages)
#If length is even then median is middle element else average of two middle elements
if(length%2==0):
    print("Median is",ages[length//2])
else:
    print("Median is",(ages[length//2]+ages[length//2+1])//2)
average=sum(ages)
print("Average of ages is",average/len(ages))
range=maximum-minimum
print("Range of ages is",range)

Ages list after sorting: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Maximum age is 26
Minimum age is 19
Ages list after appending maximum and minimum is [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 26, 19]
Median is 24
Average of ages is 22.75
Range of ages is 7
```

Question 1

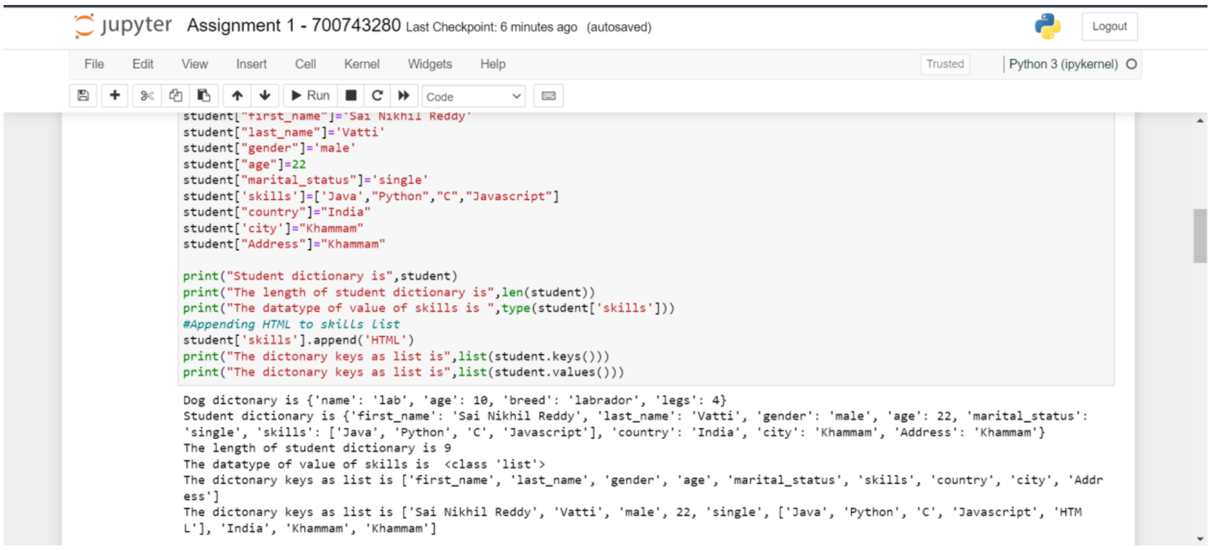
sort() is used to sort the ages and the first element will be the min and last element will be of max in the list and median is found by adding middle elements or directly taking middle element based on length and range is found using maximum-minimum


jupyter
Assignment 1 - 700743280 Last Checkpoint: 6 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [57]: #Question 2
         dog={} #Empty Dictonary
         dog['name']='lab'
         dog['age']=10
         dog['breed']='labrador'
         dog['legs']=4
         print("Dog dictonary is",dog)

         #Empty student dict
         student={}
         #Adding student details to dict
         student["first_name"]='Sai Nikhil Reddy'
         student["last_name"]='Vatti'
         student["gender"]='male'
         student["age"]=22
         student["marital_status"]='single'
         student['skills']=['Java',"Python","C","Javascript"]
         student["country"]="India"
         student['city']="Khammam"
         student["Address"]="Khammam"

         print("Student dictionary is",student)
         print("The length of student dictionary is",len(student))
         print("The datatype of value of skills is ",type(student['skills']))
         #Appending HTML to skills list
         student['skills'].append('HTML')
```


jupyter Assignment 1 - 700743280 Last Checkpoint: 6 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

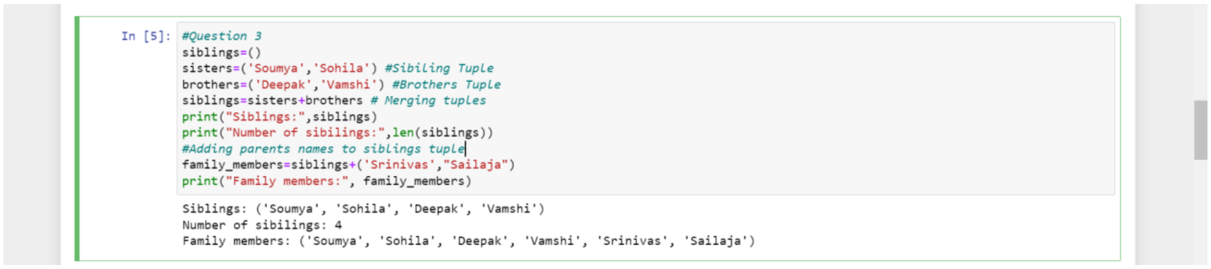
```
student["first_name"]='Sai Nikhil Reddy'
student["last_name"]='Vatti'
student["gender"]='male'
student["age"]=22
student["marital_status"]='single'
student['skills']=['Java',"Python","C","Javascript"]
student["country"]="India"
student['city']="Khammam"
student["Address"]="Khammam"

print("Student dictionary is",student)
print("The length of student dictionary is",len(student))
print("The datatype of value of skills is ",type(student['skills']))
#Appending HTML to skills list
student['skills'].append('HTML')
print("The dictonary keys as list is",list(student.keys()))
print("The dictonary keys as list is",list(student.values()))
```

```
Dog dictonary is {'name': 'lab', 'age': 10, 'breed': 'labrador', 'legs': 4}
Student dictionary is {'first_name': 'Sai Nikhil Reddy', 'last_name': 'Vatti', 'gender': 'male', 'age': 22, 'marital_status':
'single', 'skills': ['Java', 'Python', 'C', 'Javascript'], 'country': 'India', 'city': 'Khammam', 'Address': 'Khammam'}
The length of student dictionary is 9
The datatype of value of skills is  <class 'list'>
The dictonary keys as list is ['first_name', 'last_name', 'gender', 'age', 'marital_status', 'skills', 'country', 'city', 'Addr
ess']
The dictonary keys as list is ['Sai Nikhil Reddy', 'Vatti', 'male', 22, 'single', ['Java', 'Python', 'C', 'Javascript', 'HTM
L'], 'India', 'Khammam', 'Khammam']
```

Question 2

Length of dictionary is found using len() method and keys set can be found by using keys() method and values set can be found using values() method and these can be converted to list using list()

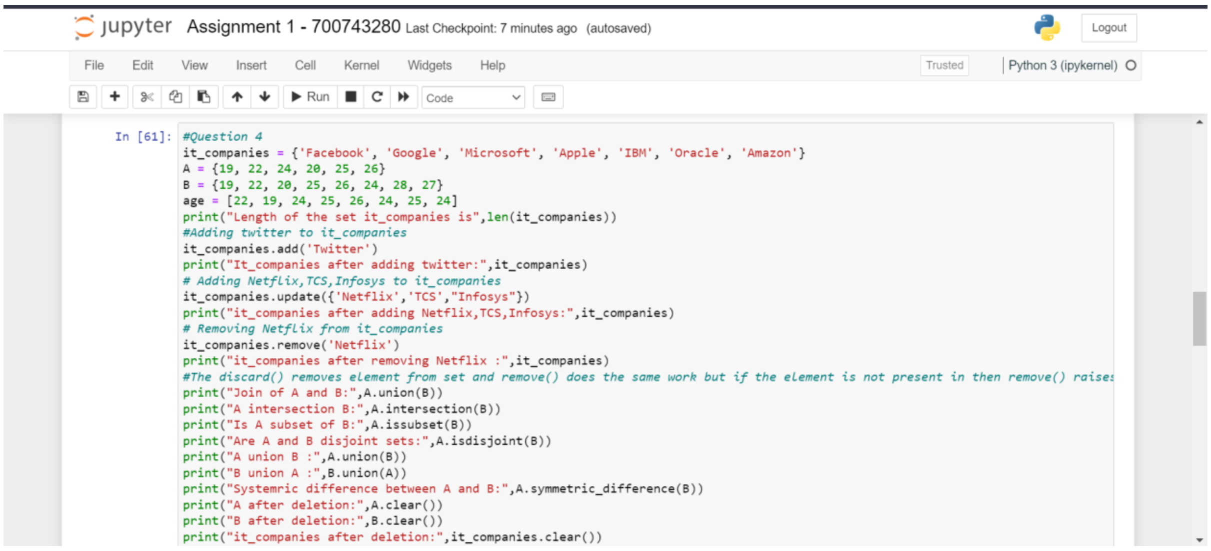


```
In [5]: #Question 3
        siblings=()
        sisters=('Soumya','Sohila') #Sibiling Tuple
        brothers=('Deepak','Vamshi') #Brothers Tuple
        siblings=sisters+brothers # Merging tuples
        print("Siblings:",siblings)
        print("Number of sibilings:",len(siblings))
        #Adding parents names to siblings tuple
        family_members=siblings+('Srinivas',"Sailaja")
        print("Family members:", family_members)
```

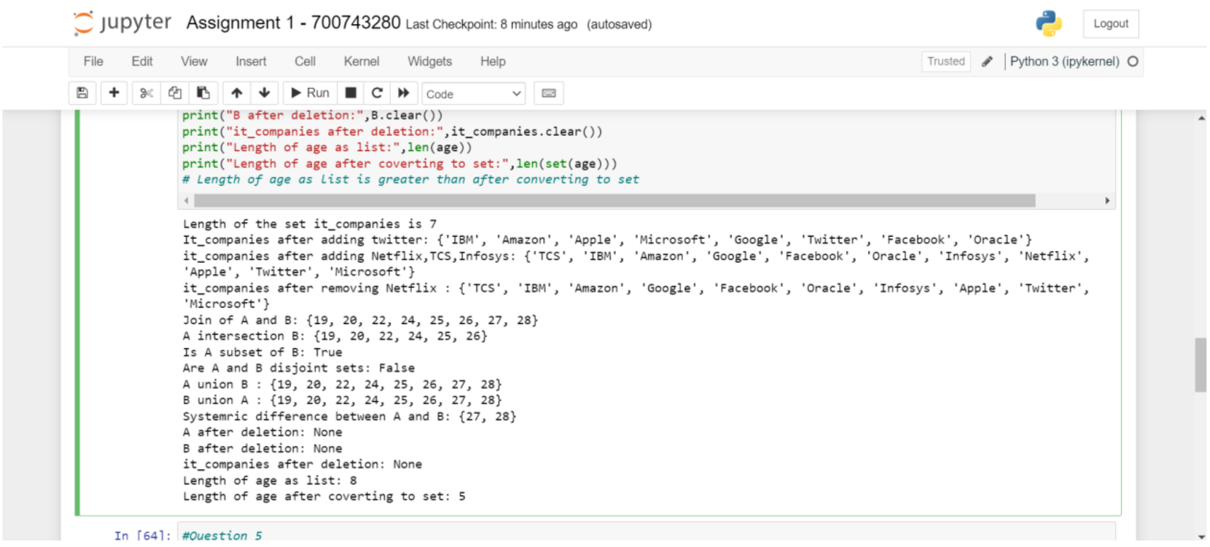
```
Siblings: ('Soumya', 'Sohila', 'Deepak', 'Vamshi')
Number of sibilings: 4
Family members: ('Soumya', 'Sohila', 'Deepak', 'Vamshi', 'Srinivas', 'Sailaja')
```

Question 3

Siblings and brothers tuples can be concatenated by using + operator.


jupyter Assignment 1 - 700743280 Last Checkpoint: 7 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [61]: #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]
         print("Length of the set it_companies is",len(it_companies))
         #Adding twitter to it_companies
         it_companies.add('Twitter')
         print("It_companies after adding twitter:",it_companies)
         # Adding Netflix,TCS,Infosys to it_companies
         it_companies.update({'Netflix','TCS',"Infosys"})
         print("it_companies after adding Netflix,TCS,Infosys:",it_companies)
         # Removing Netflix from it_companies
         it_companies.remove('Netflix')
         print("it_companies after removing Netflix :",it_companies)
         #The discard() removes element from set and remove() does the same work but if the element is not present in then remove() raises
         print("Join of A and B:",A.union(B))
         print("A intersection B:",A.intersection(B))
         print("Is A subset of B:",A.issubset(B))
         print("Are A and B disjoint sets:",A.isdisjoint(B))
         print("A union B :",A.union(B))
         print("B union A :",B.union(A))
         print("Systemric difference between A and B:",A.symmetric_difference(B))
         print("A after deletion:",A.clear())
         print("B after deletion:",B.clear())
         print("it_companies after deletion:",it_companies.clear())
```


jupyter Assignment 1 - 700743280 Last Checkpoint: 8 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
print("B after deletion:",B.clear())
print("it_companies after deletion:",it_companies.clear())
print("Length of age as list:",len(age))
print("Length of age after coverting to set:",len(set(age)))
# Length of age as list is greater than after converting to set
```

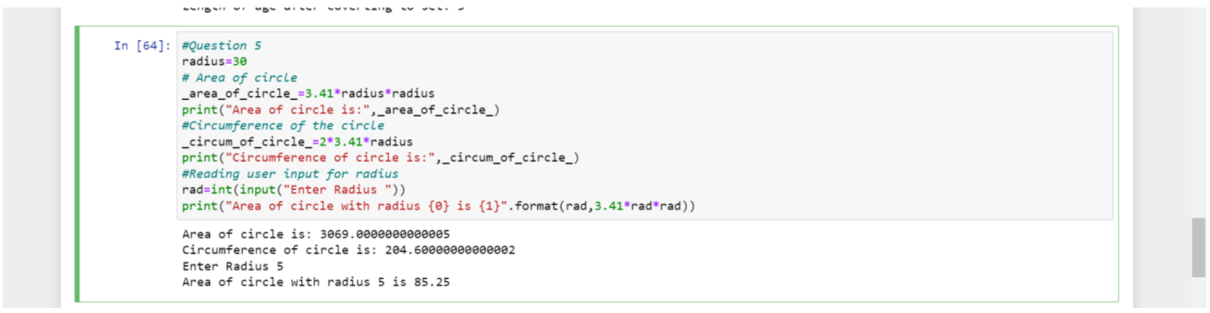
```
Length of the set it_companies is 7
It_companies after adding twitter: {'IBM', 'Amazon', 'Apple', 'Microsoft', 'Google', 'Twitter', 'Facebook', 'Oracle'}
it_companies after adding Netflix,TCS,Infosys: {'TCS', 'IBM', 'Amazon', 'Google', 'Facebook', 'Oracle', 'Infosys', 'Netflix',
'Apple', 'Twitter', 'Microsoft'}
it_companies after removing Netflix : {'TCS', 'IBM', 'Amazon', 'Google', 'Facebook', 'Oracle', 'Infosys', 'Apple', 'Twitter',
'Microsoft'}
Join of A and B: {19, 20, 22, 24, 25, 26, 27, 28}
A intersection B: {19, 20, 22, 24, 25, 26}
Is A subset of B: True
Are A and B disjoint sets: False
A union B : {19, 20, 22, 24, 25, 26, 27, 28}
B union A : {19, 20, 22, 24, 25, 26, 27, 28}
Systemric difference between A and B: {27, 28}
A after deletion: None
B after deletion: None
it_companies after deletion: None
Length of age as list: 8
Length of age after coverting to set: 5
```

```
In [64]: #Question 5
```

Question 4

Used union() method to find union of A and B sets and intersection() to find intersection() to find intersection of sets and issubset() to find whether it is a subset and symmetric_difference() is used to find symmetric difference between sets.

Difference between remove() and discard(): Both the methods remove an element from set but if the element is not present the remove() raises an error but discard() does not.

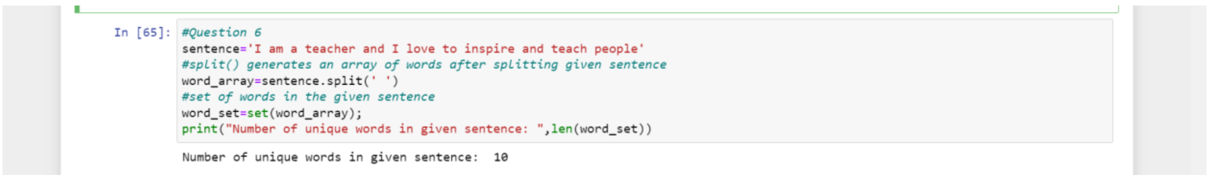


```
In [64]: #Question 5
radius=30
# Area of circle
_area_of_circle_=3.41*radius*radius
print("Area of circle is:",_area_of_circle_)
#Circumference of the circle
_circum_of_circle_=2*3.41*radius
print("Circumference of circle is:",_circum_of_circle_)
#Reading user input for radius
rad=int(input("Enter Radius "))
print("Area of circle with radius {0} is {1}".format(rad,3.41*rad*rad))
```

```
Area of circle is: 3069.0000000000005
Circumference of circle is: 204.60000000000002
Enter Radius 5
Area of circle with radius 5 is 85.25
```

Question 5

input() method is used to read the input from user and this method by default return an string and this string is converted to integer using int() and this value is stored in rad variable.

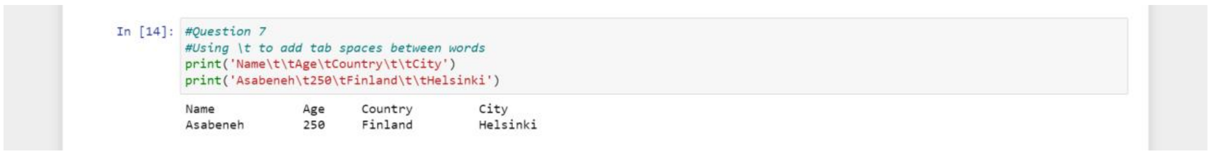


```
In [65]: #Question 6
sentence='I am a teacher and I love to inspire and teach people'
#split() generates an array of words after splitting given sentence
word_array=sentence.split(' ')
#set of words in the given sentence
word_set=set(word_array);
print("Number of unique words in given sentence: ",len(word_set))
```

```
Number of unique words in given sentence:  10
```

Question 6

split(“ ”) is used to split the given sentence by space and this method return an word array this word array is then converted to set by using the set() method and it can be observed that the length after converting to set is than that of the original list because the set method removes the duplicates and returns a set.

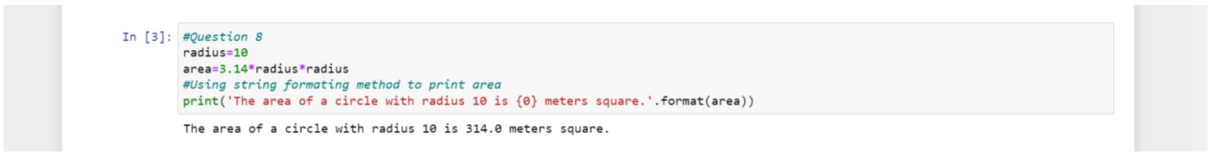


```
In [14]: #Question 7
         #Using \t to add tab spaces between words
         print('Name\t\tAge\tCountry\t\tCity')
         print('Asabeneh\t250\tFinland\t\tHelsinki')

         Name            Age     Country         City
         Asabeneh        250     Finland         Helsinki
```

Question 7

\t is used to insert tab spaces in the given sentence.

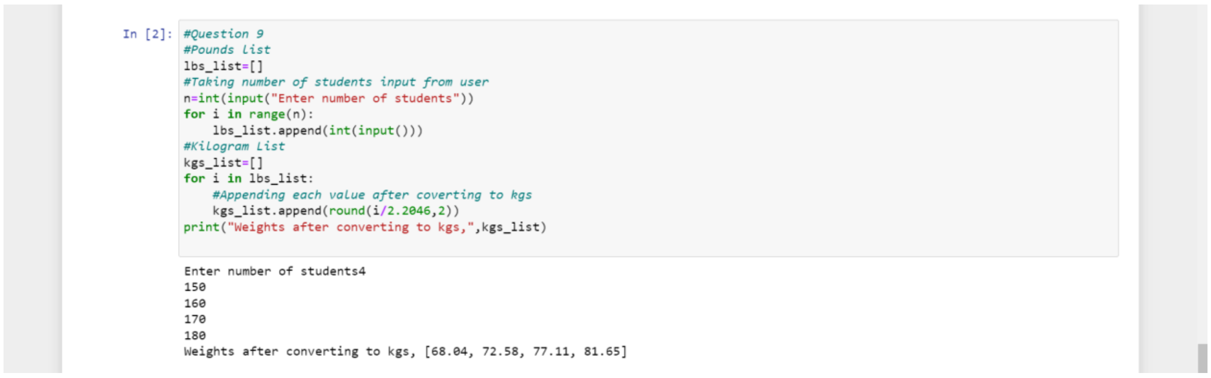


```
In [3]: #Question 8
        radius=10
        area=3.14*radius*radius
        #Using string formating method to print area
        print('The area of a circle with radius 10 is {0} meters square.'.format(area))

        The area of a circle with radius 10 is 314.0 meters square.
```

Question 8

format() method is used to format the string and replacing the {0} with radius given by the user and {1} is replace by the calculated area.



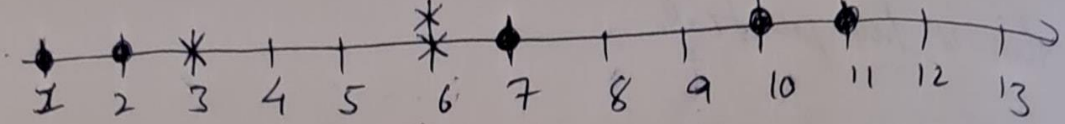
```
In [2]: #Question 9
        #Pounds List
        lbs_list=[]
        #Taking number of students input from user
        n=int(input("Enter number of students"))
        for i in range(n):
            lbs_list.append(int(input()))
        #Kilogram List
        kgs_list=[]
        for i in lbs_list:
            #Appending each value after coverting to kgs
            kgs_list.append(round(i/2.2046,2))
        print("Weights after converting to kgs,",kgs_list)

        Enter number of students4
        150
        160
        170
        180
        Weights after converting to kgs, [68.04, 72.58, 77.11, 81.65]
```

Question 9

Empty pounds list and created and number of students is taken as input from user and each item in the list is taken as input and pushed into pounds list and each element in this list is divided by 2.2046 and pushed into kilogram list.

Question 10:-



Let us split the data into two equal parts for training & testing

Xtraining : [1, 3, 6, 10]

~~[illegible]~~ Ytraining : ['dot', 'cross, 'cross', 'dot']

Xtesting : [2, 6, 7, 11]

Ytesting : ['dot', 'cross', 'dot', 'dot']

Confusion matrix :-

$$\begin{array}{c|c|c|}
 & O & X \\
\hline
O & TN & FP \\
\hline
X & FN & TP. \\
\hline
\end{array}$$

Alogrithm for 3neighbours :-

~~[illegible]~~ The nearest neighbours for 2 are [1, 3, 6] which have the values of ['dot', 'cross', 'cross') so the predicted value for 2 is 'cross'

so this is a 'False positive'

ii) The nearest neighbour for 6 are [6, 7, 3) and these have values of ('dot', 'cross', 'dot', 'cross']

so the predicted value for 6 is 'cross'

so this is a True positive

iii) The nearest neighbour => 7 => [6, 6, 10] => values are ['cross', 'cross', 'dot'],

Predicted value for 7 is 'cross'

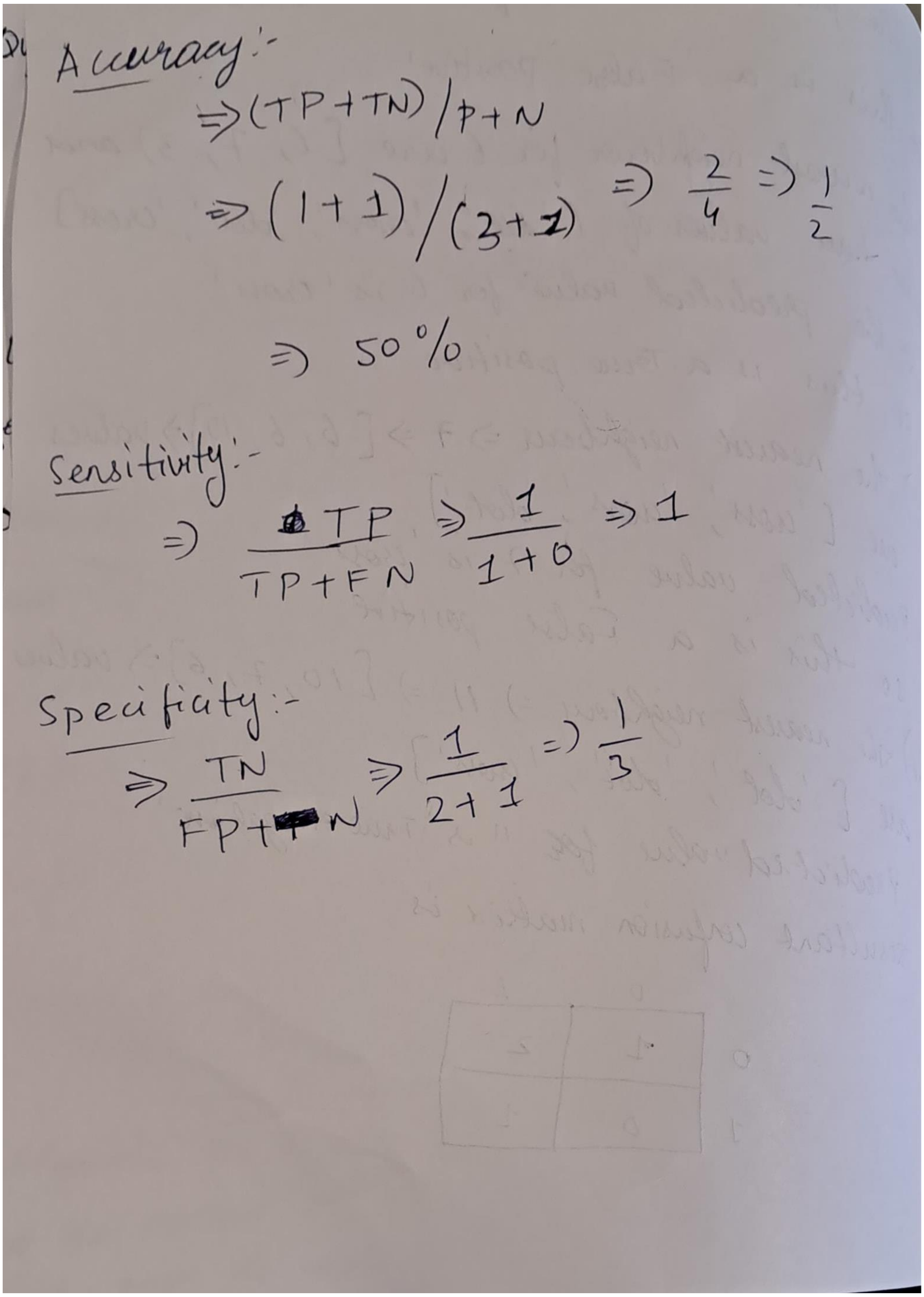
so this is a False positive

iv) The nearest neighbour => 11 => [10, 7, 6] => values are ['dot', 'dot', 'cross']

Predicted value for 11 is 'True Negative'

Resultant confusion matrix is

$$\begin{array}{c|cc} & 0 & 1 \\ \hline 0 & 1 & 2 \\ 1 & 0 & 1 \end{array}$$



Accuracy:-

$\Rightarrow (TP+TN)/P+N$

$\Rightarrow (1+1)/(3+1) \Rightarrow \frac{2}{4} \Rightarrow \frac{1}{2}$

$\Rightarrow 50\%$

Sensitivity:-

$\Rightarrow \frac{TP}{TP+FN} \Rightarrow \frac{1}{1+0} \Rightarrow 1$

Specificity:-

$\Rightarrow \frac{TN}{FP+N} \Rightarrow \frac{1}{2+1} \Rightarrow \frac{1}{3}$

Question 10

Github link: https://github.com/sainikhil3280/ML-Assignment-1

Video Link: https://drive.google.com/drive/u/1/folders/1BqRGZ-k3ImPzhrAU7HqO_lNrR4PKk5xV