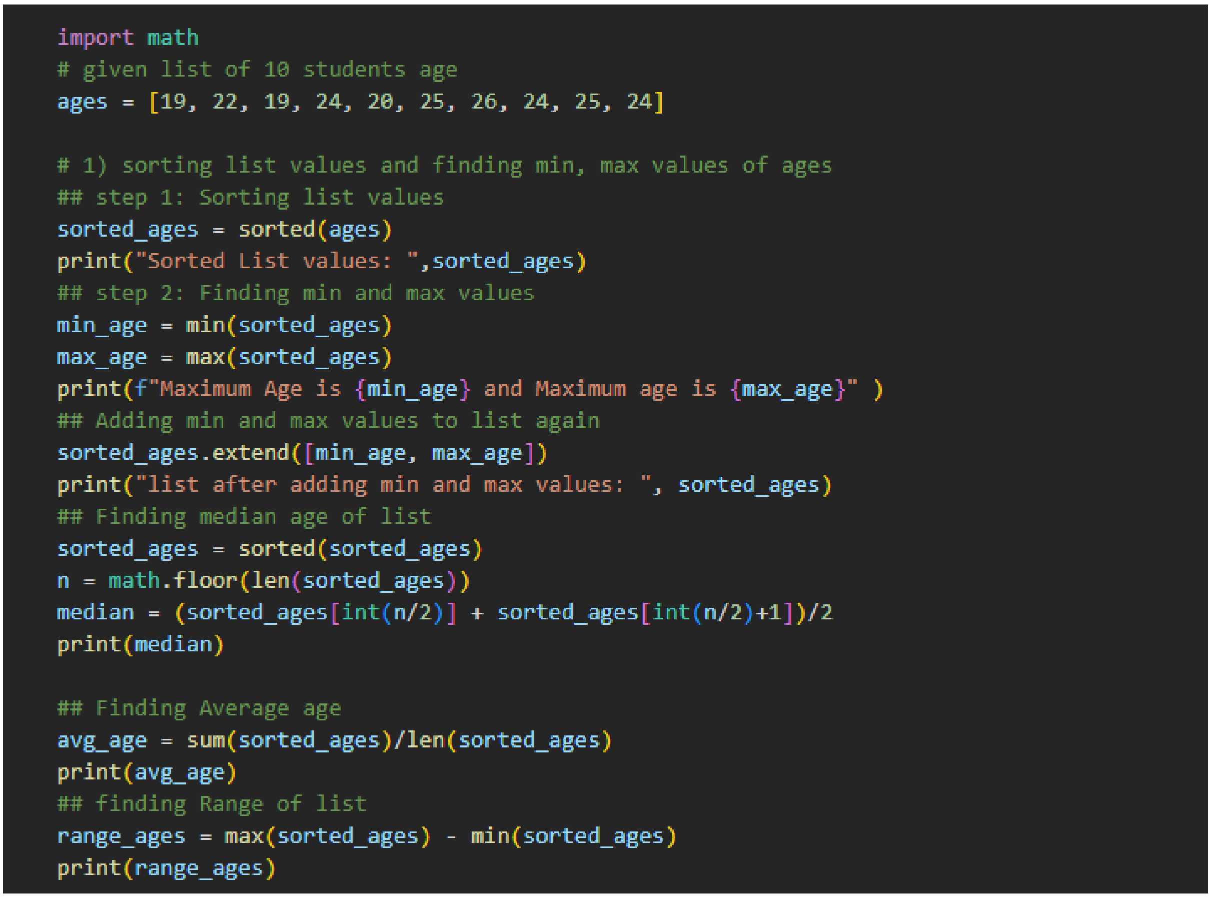
Machine Learning Assignment 1
Laxma Reddy Nalla
700732071

Video Link: https://youtu.be/qxOLxUhCPS4

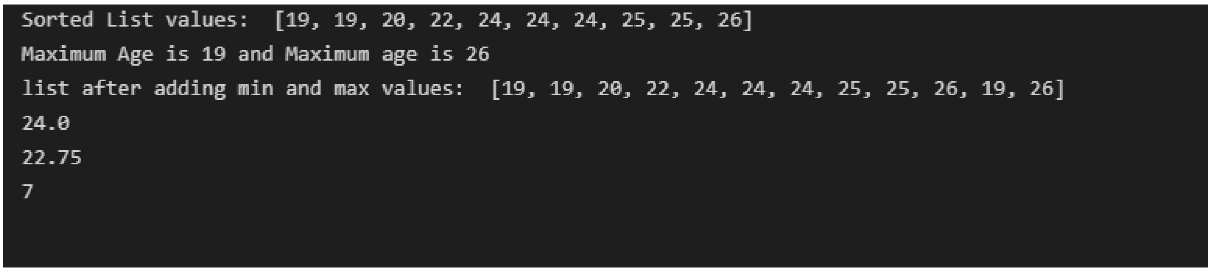
Github Link: [CS-5710/Assigment-01 at master · LaxmaReddy-Nalla/CS-5710 (github.com)]()

Question 1:
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
For this array find
Sort the array and finding max and min values
Adding min and max values to sorted array
Finding median age
Finding average age
Finding range of values
Code:



```
import math
# given list of 10 students age
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

# 1) sorting list values and finding min, max values of ages
## step 1: Sorting list values
sorted_ages = sorted(ages)
print("Sorted List values: ",sorted_ages)
## step 2: Finding min and max values
min_age = min(sorted_ages)
max_age = max(sorted_ages)
print(f"Maximum Age is {min_age} and Maximum age is {max_age}" )
## Adding min and max values to list again
sorted_ages.extend([min_age, max_age])
print("list after adding min and max values: ", sorted_ages)
## Finding median age of list
sorted_ages = sorted(sorted_ages)
n = math.floor(len(sorted_ages))
median = (sorted_ages[int(n/2)] + sorted_ages[int(n/2)+1])/2
print(median)

## Finding Average age
avg_age = sum(sorted_ages)/len(sorted_ages)
print(avg_age)
## finding Range of list
range_ages = max(sorted_ages) - min(sorted_ages)
print(range_ages)
```

Result:

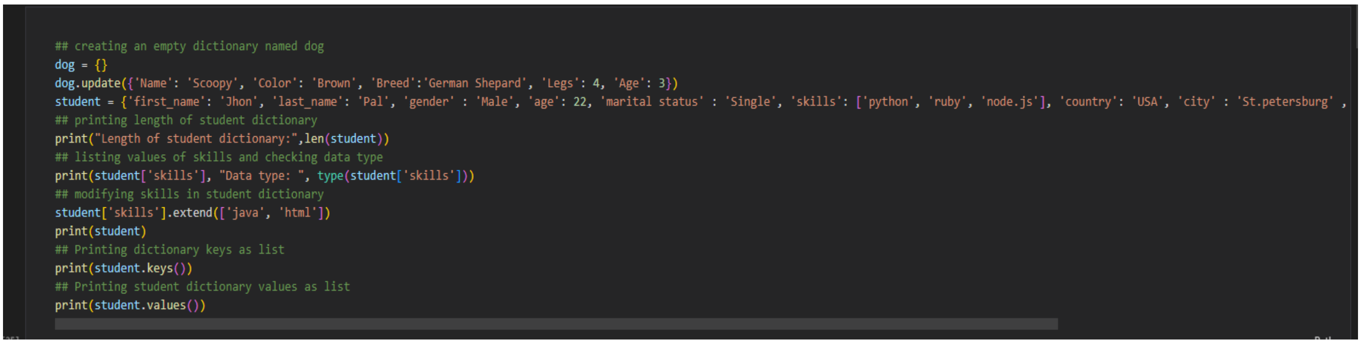


```
Sorted List values:  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Maximum Age is 19 and Maximum age is 26
list after adding min and max values:  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
24.0
22.75
7
```

Question 2:
Here in the 2nd question I created an empty dict named dog and updated values such as name, color, breed, legs, age. Created a new dictionary named student and added values such as first_name, last_name, gender, age, marital status, skills, country, city and address.

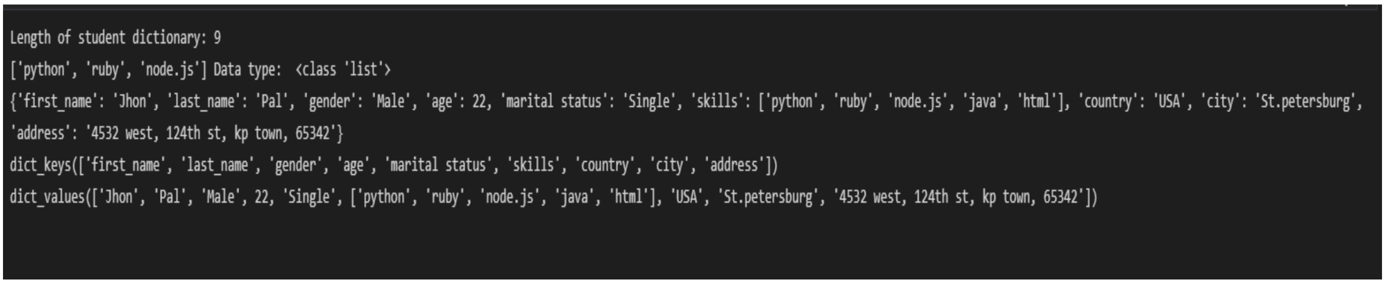
printed length of student dict length, get values of skills and type of skills
Then modified the skills list and added two more skills to the list.
Printed the values and keys list from the dictionary.

Code:



```
## creating an empty dictionary named dog
dog = {}
dog.update({'Name': 'Scoopy', 'Color': 'Brown', 'Breed':'German Shepard', 'Legs': 4, 'Age': 3})
student = {'first_name': 'Jhon', 'last_name': 'Pal', 'gender' : 'Male', 'age': 22, 'marital status' : 'Single', 'skills': ['python', 'ruby', 'node.js'], 'country': 'USA', 'city' : 'St.petersburg' ,
## printing length of student dictionary
print("Length of student dictionary:",len(student))
## listing values of skills and checking data type
print(student['skills'], "Data type: ", type(student['skills']))
## modifying skills in student dictionary
student['skills'].extend(['java', 'html'])
print(student)
## Printing dictionary keys as list
print(student.keys())
## Printing student dictionary values as list
print(student.values())
```

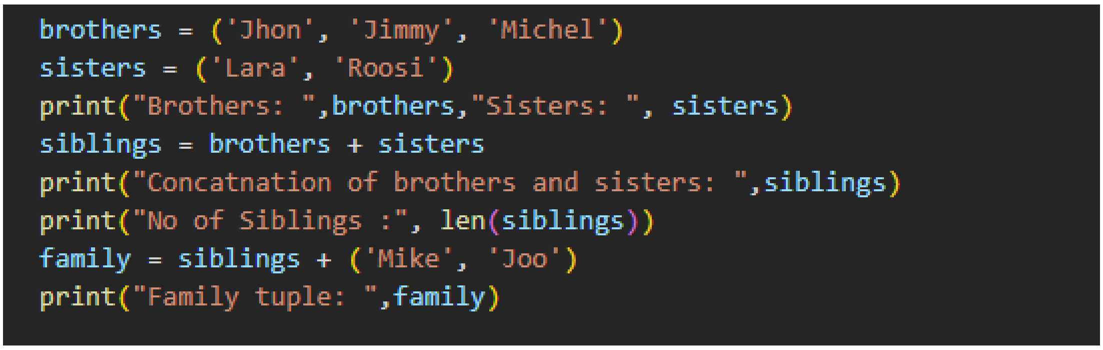
Results:



```
Length of student dictionary: 9
['python', 'ruby', 'node.js'] Data type:  <class 'list'>
{'first_name': 'Jhon', 'last_name': 'Pal', 'gender': 'Male', 'age': 22, 'marital status': 'Single', 'skills': ['python', 'ruby', 'node.js', 'java', 'html'], 'country': 'USA', 'city': 'St.petersburg',
'address': '4532 west, 124th st, kp town, 65342'}
dict_keys(['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'address'])
dict_values(['Jhon', 'Pal', 'Male', 22, 'Single', ['python', 'ruby', 'node.js', 'java', 'html'], 'USA', 'St.petersburg', '4532 west, 124th st, kp town, 65342'])
```

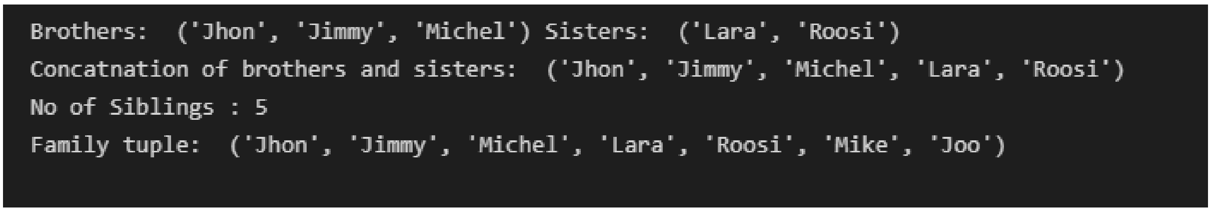
Question3 :
Created two tuples named brothers and sisters then joined those tuples as siblings then found length of tuple and created another tuple Family and added siblings and father and mother.

Code:



```
brothers = ('Jhon', 'Jimmy', 'Michel')
sisters = ('Lara', 'Roosi')
print("Brothers: ",brothers,"Sisters: ", sisters)
siblings = brothers + sisters
print("Concatnation of brothers and sisters: ",siblings)
print("No of Siblings :", len(siblings))
family = siblings + ('Mike', 'Joo')
print("Family tuple: ",family)
```

Results:



```
Brothers:  ('Jhon', 'Jimmy', 'Michel') Sisters:  ('Lara', 'Roosi')
Concatnation of brothers and sisters:  ('Jhon', 'Jimmy', 'Michel', 'Lara', 'Roosi')
No of Siblings : 5
Family tuple:  ('Jhon', 'Jimmy', 'Michel', 'Lara', 'Roosi', 'Mike', 'Joo')
```

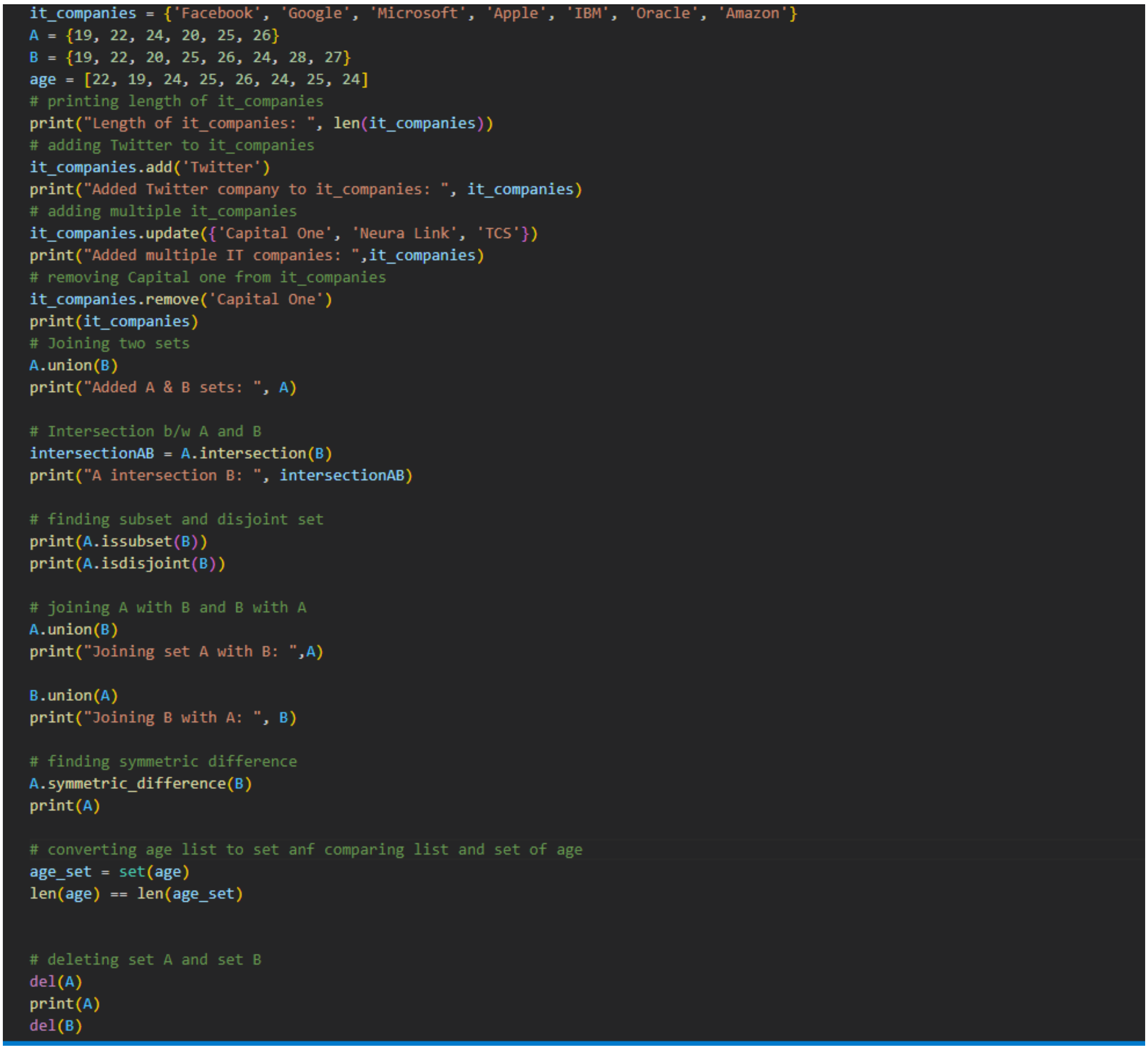
Question 4:

What is the difference between remove and discord?

The remove method raises an error if a specified element is not found. discord method doesn’t raise an error if an element is not found in set.

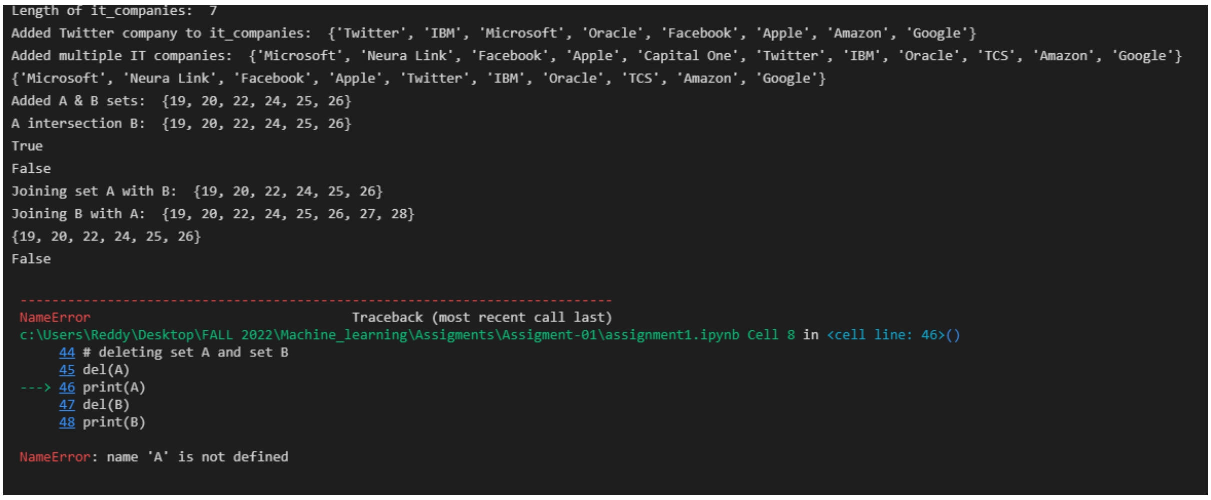
Find the length of the set it_companies • Add 'Twitter' to it_companies
• Insert multiple IT companies at once to the set it_companies
• Remove one of the companies from the set it_companies
• Join A and B
• Find A intersection B

- 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.



```python
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]
# printing length of it_companies
print("Length of it_companies: ", len(it_companies))
# adding Twitter to it_companies
it_companies.add('Twitter')
print("Added Twitter company to it_companies: ", it_companies)
# adding multiple it_companies
it_companies.update({'Capital One', 'Neura Link', 'TCS'})
print("Added multiple IT companies: ",it_companies)
# removing Capital one from it_companies
it_companies.remove('Capital One')
print(it_companies)
# Joining two sets
A.union(B)
print("Added A & B sets: ", A)

# Intersection b/w A and B
intersectionAB = A.intersection(B)
print("A intersection B: ", intersectionAB)

# finding subset and disjoint set
print(A.issubset(B))
print(A.isdisjoint(B))

# joining A with B and B with A
A.union(B)
print("Joining set A with B: ",A)

B.union(A)
print("Joining B with A: ", B)

# finding symmetric difference
A.symmetric_difference(B)
print(A)

# converting age list to set anf comparing list and set of age
age_set = set(age)
len(age) == len(age_set)


# deleting set A and set B
del(A)
print(A)
del(B)
```

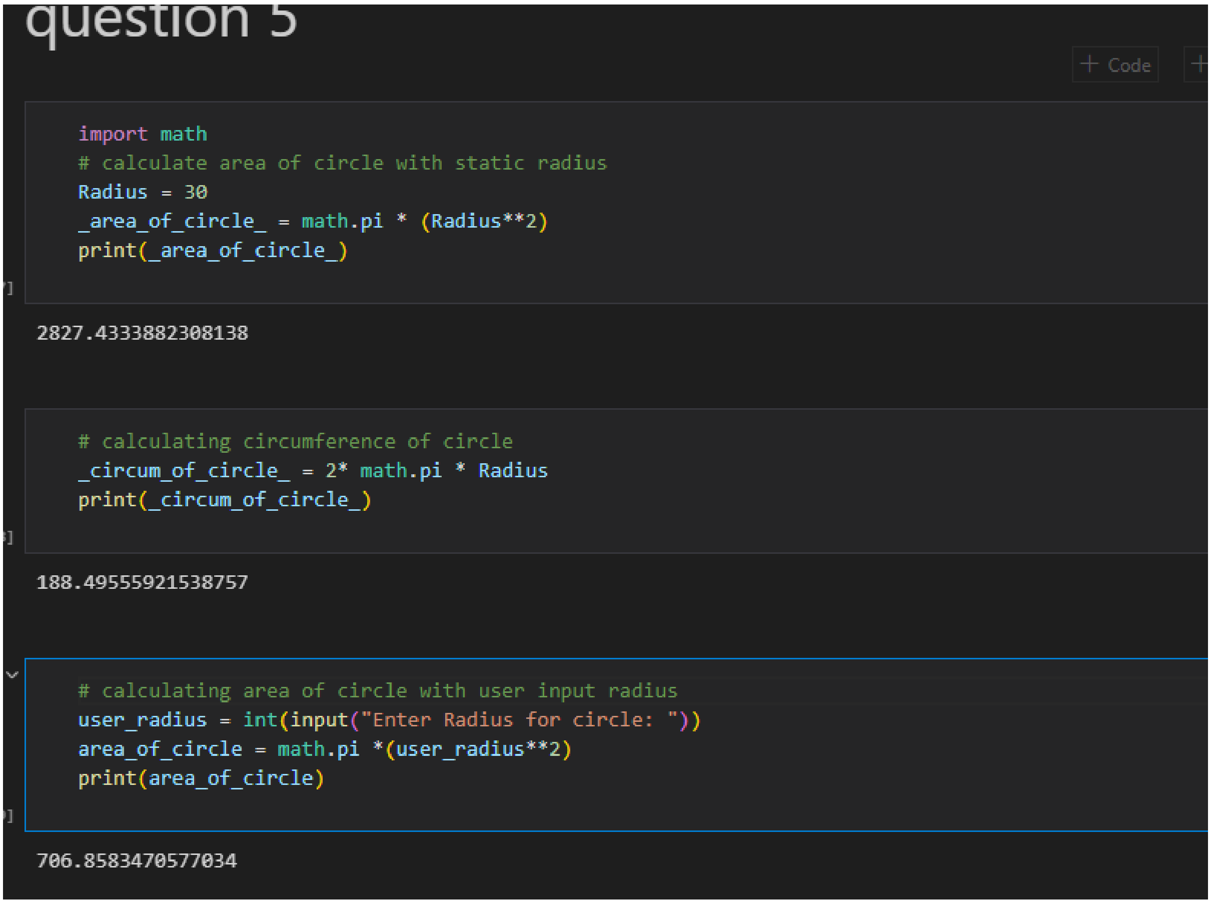
Results:



```
Length of it_companies:  7
Added Twitter company to it_companies:  {'Twitter', 'IBM', 'Microsoft', 'Oracle', 'Facebook', 'Apple', 'Amazon', 'Google'}
Added multiple IT companies:  {'Microsoft', 'Neura Link', 'Facebook', 'Apple', 'Capital One', 'Twitter', 'IBM', 'Oracle', 'TCS', 'Amazon', 'Google'}
{'Microsoft', 'Neura Link', 'Facebook', 'Apple', 'Twitter', 'IBM', 'Oracle', 'TCS', 'Amazon', 'Google'}
Added A & B sets:  {19, 20, 22, 24, 25, 26}
A intersection B:  {19, 20, 22, 24, 25, 26}
True
False
Joining set A with B:  {19, 20, 22, 24, 25, 26}
Joining B with A:  {19, 20, 22, 24, 25, 26, 27, 28}
{19, 20, 22, 24, 25, 26}
False

 ---------------------------------------------------------------------------
 NameError                                 Traceback (most recent call last)
 c:\Users\Reddy\Desktop\FALL 2022\Machine_learning\Assigments\Assigment-01\assignment1.ipynb Cell 8 in <cell line: 46>()
      44 # deleting set A and set B
      45 del(A)
 ---> 46 print(A)
      47 del(B)
      48 print(B)

 NameError: name 'A' is not defined
```

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.



question 5

```
import math
# calculate area of circle with static radius
Radius = 30
_area_of_circle_ = math.pi * (Radius**2)
print(_area_of_circle_)
```

```
2827.4333882308138
```

```
# calculating circumference of circle
_circum_of_circle_ = 2* math.pi * Radius
print(_circum_of_circle_)
```

```
188.49555921538757
```

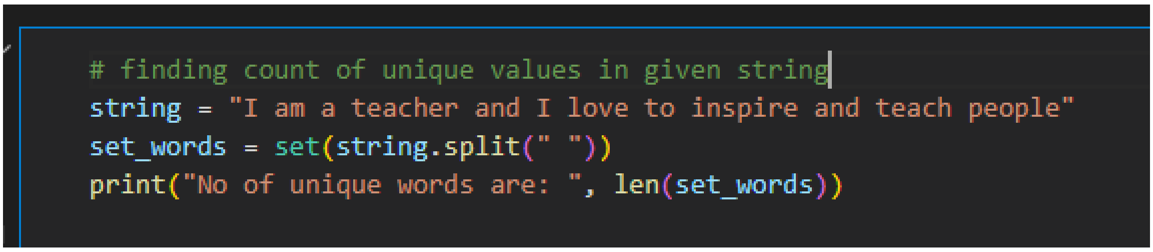
```
# calculating area of circle with user input radius
user_radius = int(input("Enter Radius for circle: "))
area_of_circle = math.pi *(user_radius**2)
print(area_of_circle)
```

```
706.8583470577034
```

Question 6:

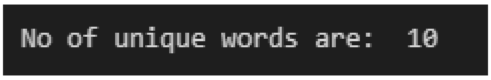
“I am a teacher and I love to inspire and teach people”
• How many unique words have been used in the sentence? Use the split methods and set to get the unique words

Using split and set found unique values



```
# finding count of unique values in given string
string = "I am a teacher and I love to inspire and teach people"
set_words = set(string.split(" "))
print("No of unique words are: ", len(set_words))
```

Results:



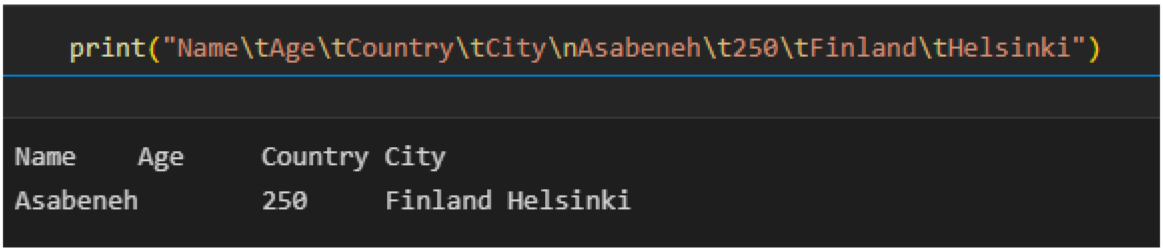
```
No of unique words are:  10
```

Question 7:

Use a tab escape sequence to get the following lines.

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

Code & Result:



```
print("Name\tAge\tCountry\tCity\nAsabeneh\t250\tFinland\tHelsinki")
```

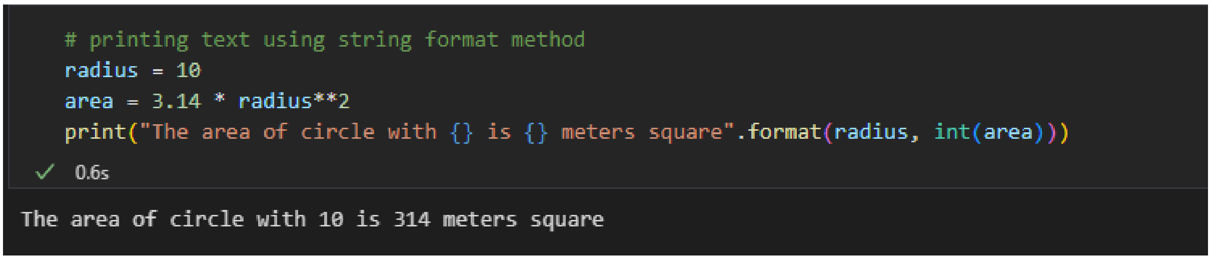
```
Name    Age     Country City
Asabeneh        250     Finland Helsinki
```

Question 8:

Use the string formatting method to display the following:

“The area of a circle with radius 10 is 314 meters square.”

Code & Result:



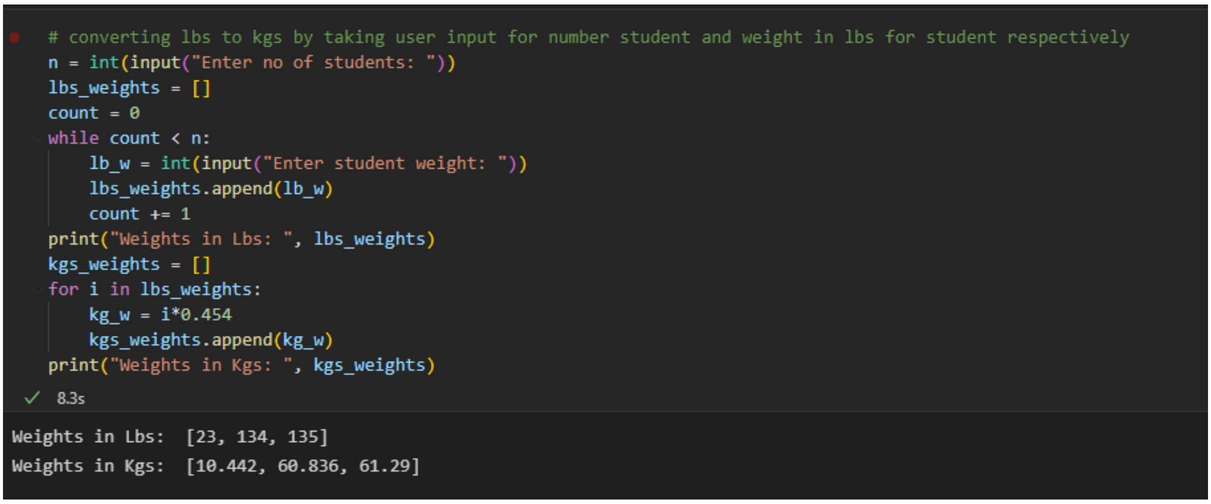
```
# printing text using string format method
radius = 10
area = 3.14 * radius**2
print("The area of circle with {} is {} meters square".format(radius, int(area)))
```

✓ 0.6s

```
The area of circle with 10 is 314 meters square
```

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)

Code & Result:



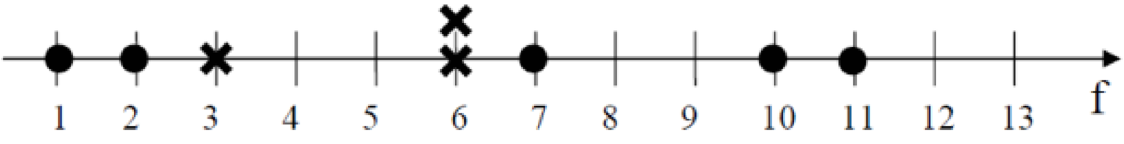
```
# converting lbs to kgs by taking user input for number student and weight in lbs for student respectively
n = int(input("Enter no of students: "))
lbs_weights = []
count = 0
while count < n:
    lb_w = int(input("Enter student weight: "))
    lbs_weights.append(lb_w)
    count += 1
print("Weights in Lbs: ", lbs_weights)
kgs_weights = []
for i in lbs_weights:
    kg_w = i*0.454
    kgs_weights.append(kg_w)
print("Weights in Kgs: ", kgs_weights)
```

✓ 8.3s

```
Weights in Lbs:  [23, 134, 135]
Weights in Kgs:  [10.442, 60.836, 61.29]
```

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.



Divide this data equally into two parts. Use the 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.

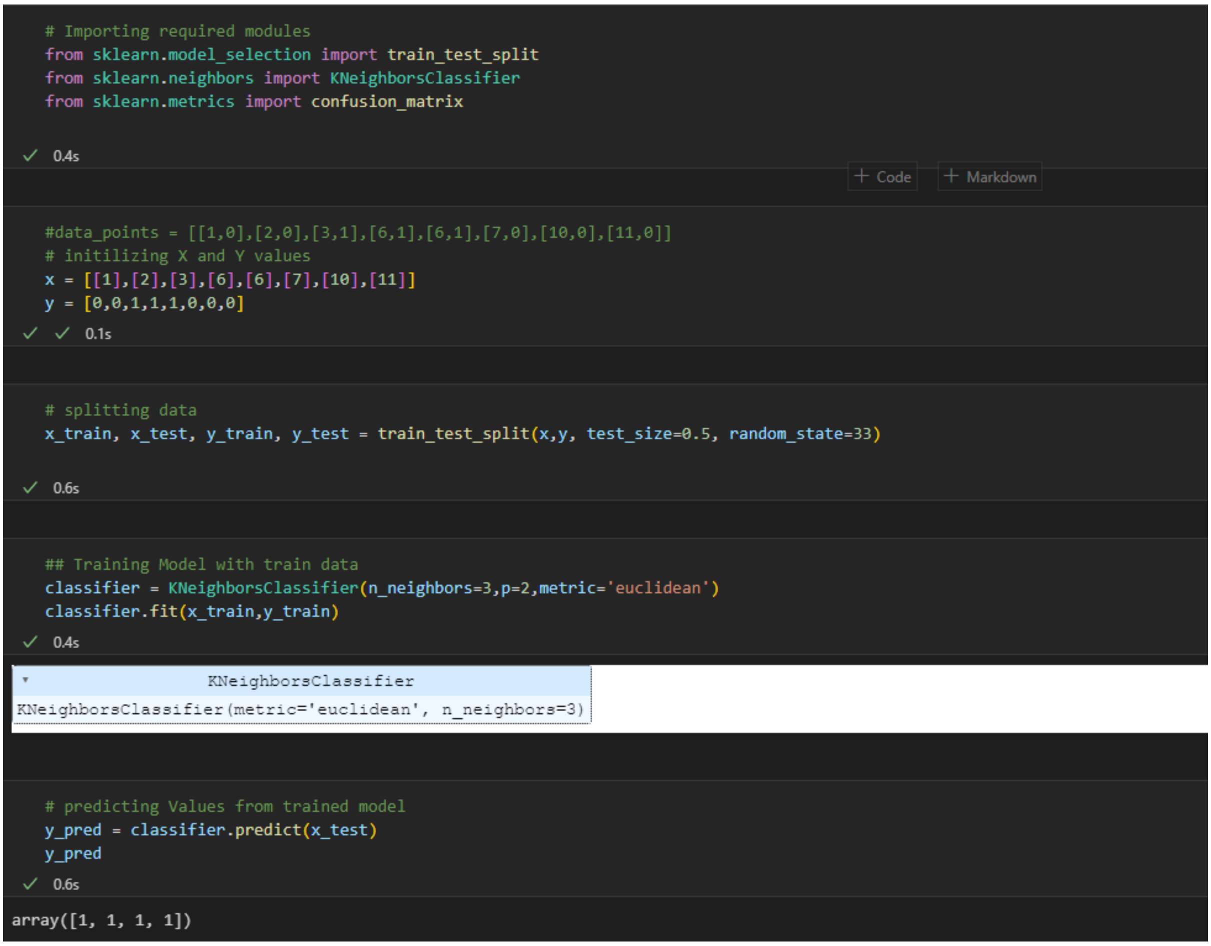
Solution:

Here in the given data set we have 2 classes. I assumed that Dot points are animal Dog(0) and cross as animal Cat(1).
So the assumed data is like:

(1,0), (2,0), (3,1), (6,1), (6,1), (7,0), (10,0), (11,0)

Now Imported required modules from sklearn such as KNN, initialized values for X and Y then splitted data into test and train set. Then Trained the model using training data with half of real data and using K as 3 and metric as euclidean distance. After training the model predicted values from the model using test data which initially splitted from the read data set.

Code:



```
# Importing required modules
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix
```

```
#data_points = [[1,0],[2,0],[3,1],[6,1],[6,1],[7,0],[10,0],[11,0]]
# initilizing X and Y values
x = [[1],[2],[3],[6],[6],[7],[10],[11]]
y = [0,0,1,1,1,0,0,0]
```

```
# splitting data
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size=0.5, random_state=33)
```

```
## Training Model with train data
classifier = KNeighborsClassifier(n_neighbors=3,p=2,metric='euclidean')
classifier.fit(x_train,y_train)
```

```
▾ KNeighborsClassifier
KNeighborsClassifier(metric='euclidean', n_neighbors=3)
```

```
# predicting Values from trained model
y_pred = classifier.predict(x_test)
y_pred
```

```
array([1, 1, 1, 1])
```

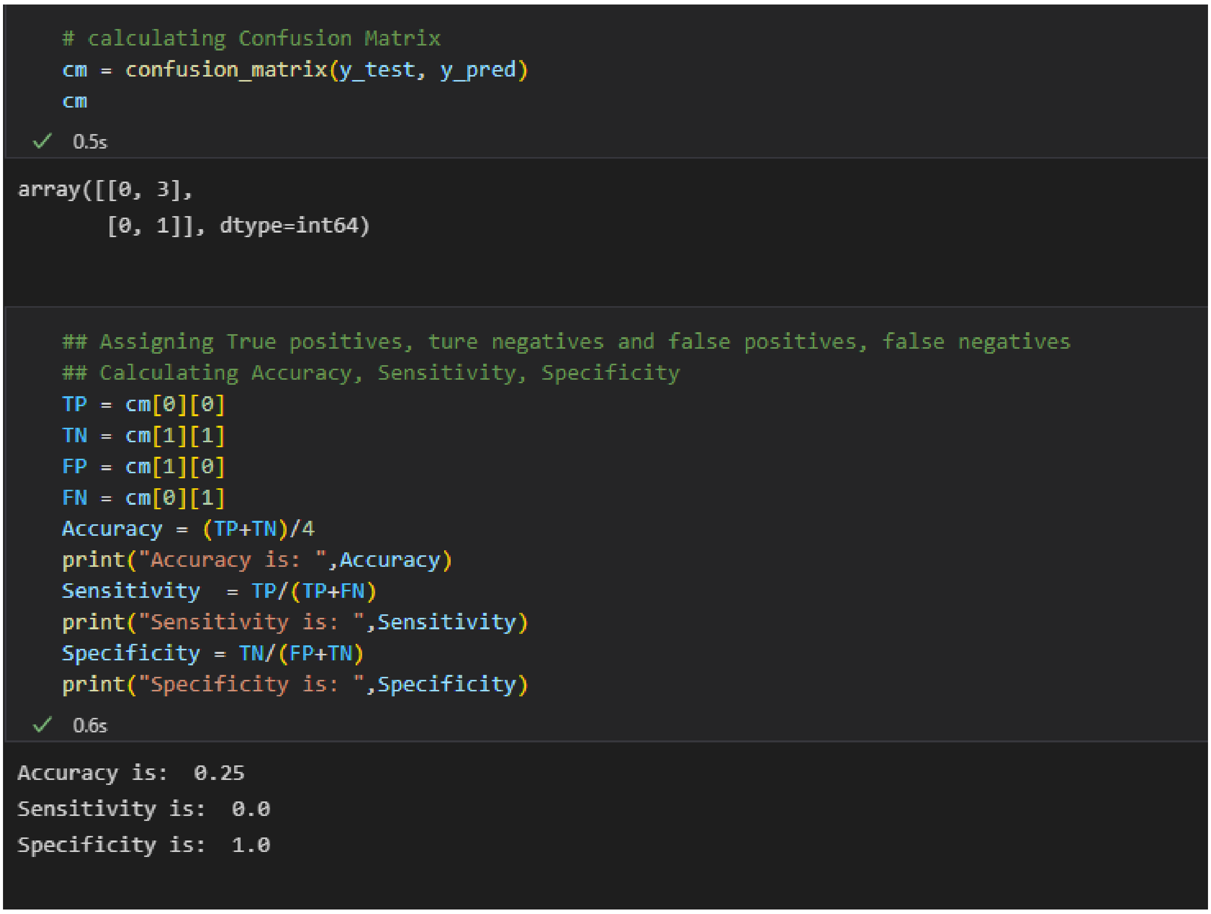
After predicting values from the test data set calculated Accuracy, sensitivity and specificity using Confusion matrix values.

Accuracy = (TP+TN)/P+N

Sensitivity = TP/(TP+FN)

Specificity = TN/(FP+TN)

Code & Result:



```python
# calculating Confusion Matrix
cm = confusion_matrix(y_test, y_pred)
cm
```

```
array([[0, 3],
       [0, 1]], dtype=int64)
```

```python
## Assigning True positives, ture negatives and false positives, false negatives
## Calculating Accuracy, Sensitivity, Specificity
TP = cm[0][0]
TN = cm[1][1]
FP = cm[1][0]
FN = cm[0][1]
Accuracy = (TP+TN)/4
print("Accuracy is: ",Accuracy)
Sensitivity  = TP/(TP+FN)
print("Sensitivity is: ",Sensitivity)
Specificity = TN/(FP+TN)
print("Specificity is: ",Specificity)
```

```
Accuracy is:  0.25
Sensitivity is:  0.0
Specificity is:  1.0
```