Machine Learning- Assignment_1

Singam Manasvi

CRN:CS 5710

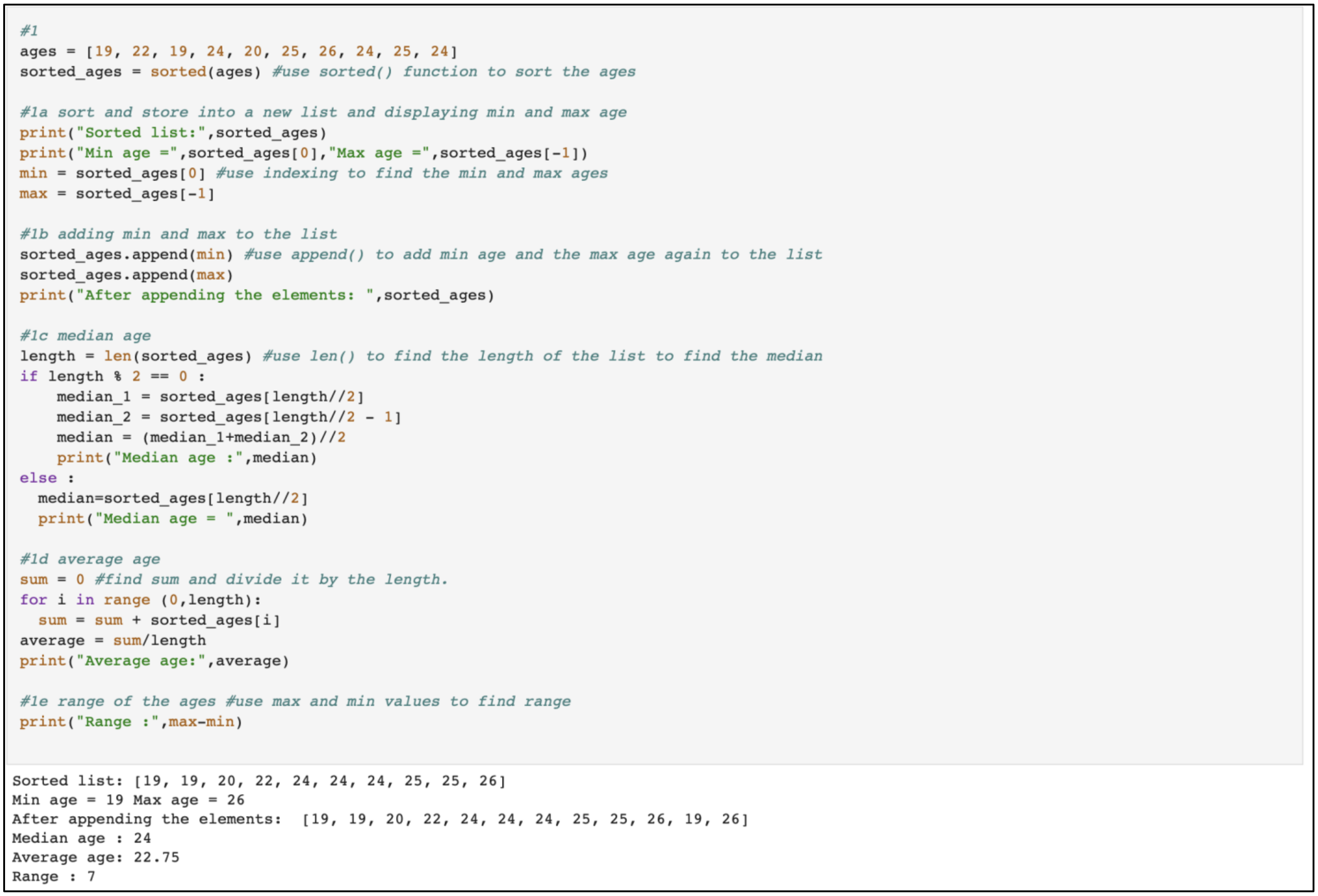
Course Id:23921

Student Id:700742501

Semester:2

GitHub Link: https://github.com/singammanasvi9440/Assignment1

1.



```
#1
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
sorted_ages = sorted(ages) #use sorted() function to sort the ages

#1a sort and store into a new list and displaying min and max age
print("Sorted list:",sorted_ages)
print("Min age =",sorted_ages[0],"Max age =",sorted_ages[-1])
min = sorted_ages[0] #use indexing to find the min and max ages
max = sorted_ages[-1]

#1b adding min and max to the list
sorted_ages.append(min) #use append() to add min age and the max age again to the list
sorted_ages.append(max)
print("After appending the elements: ",sorted_ages)

#1c median age
length = len(sorted_ages) #use len() to find the length of the list to find the median
if length % 2 == 0 :
    median_1 = sorted_ages[length//2]
    median_2 = sorted_ages[length//2 - 1]
    median = (median_1+median_2)//2
    print("Median age :",median)
else :
  median=sorted_ages[length//2]
  print("Median age = ",median)

#1d average age
sum = 0 #find sum and divide it by the length.
for i in range (0,length):
  sum = sum + sorted_ages[i]
average = sum/length
print("Average age:",average)

#1e range of the ages #use max and min values to find range
print("Range :",max-min)
```

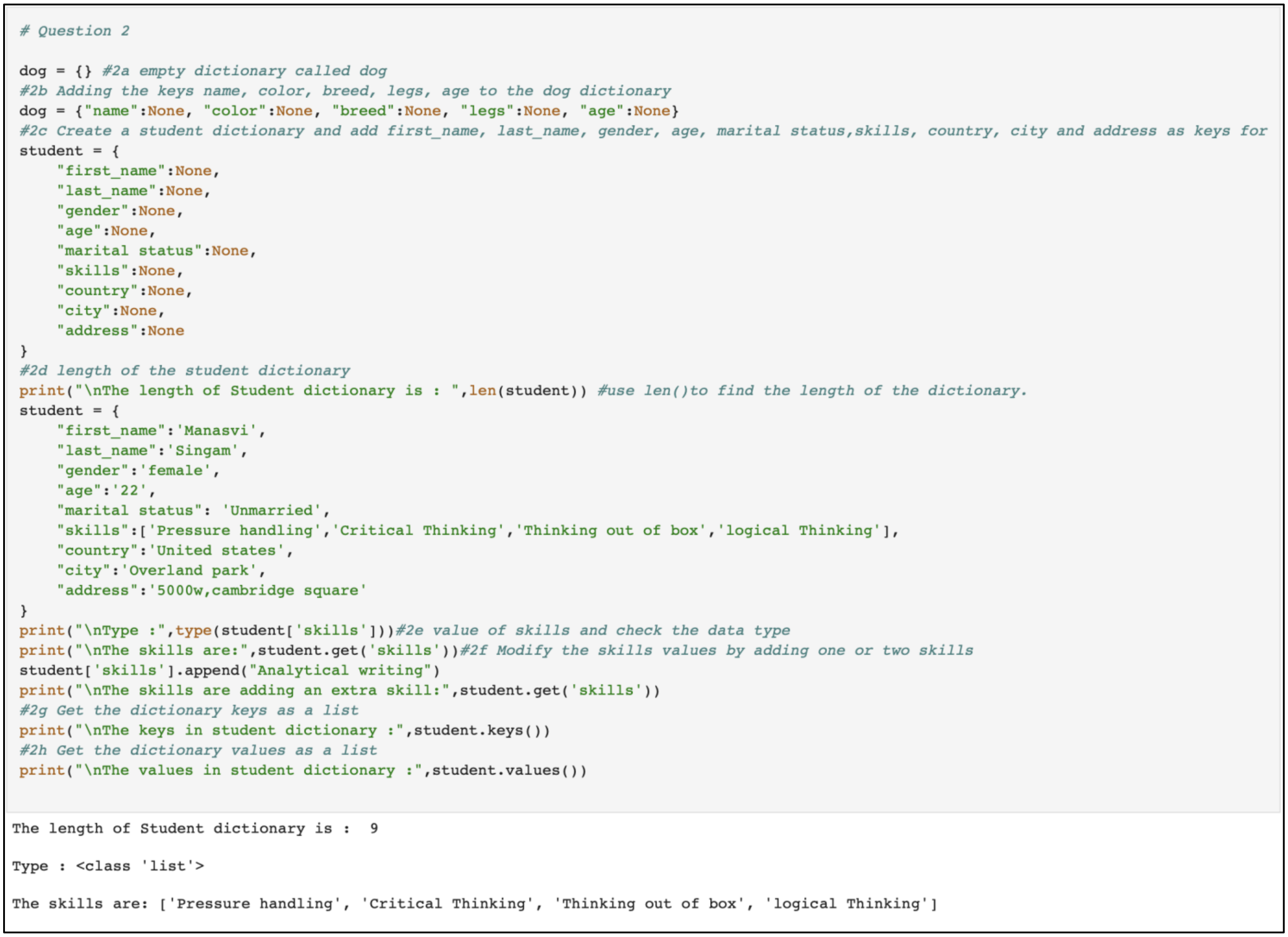
```
Sorted list: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Min age = 19 Max age = 26
After appending the elements:  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Median age : 24
Average age: 22.75
Range : 7
```

Explanation:

- Take a list and use sorted () to sort the ages.
- To add the min and max ages use indexing and then using append () add the max and min ages into the list.

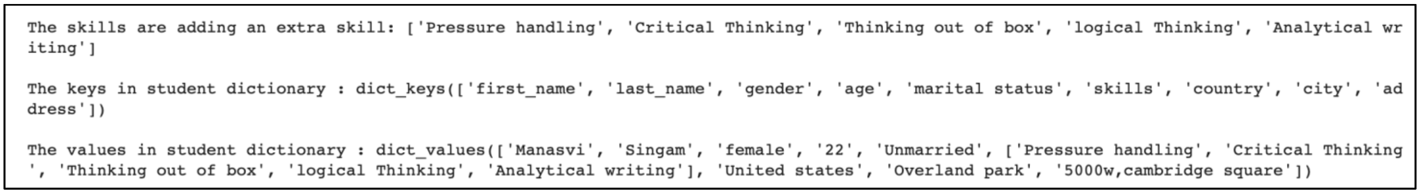
- To find median first find the length of the list using len() and divide it by 2. If the modulus answer is Zero, then median is one middle item else two middle items divided by two.
- To find the average age, calculate the sum using for loop and divide it by the length of the list.

2.



```
# Question 2

dog = {} #2a empty dictionary called dog
#2b Adding the keys name, color, breed, legs, age to the dog dictionary
dog = {"name":None, "color":None, "breed":None, "legs":None, "age":None}
#2c Create a student dictionary and add first_name, last_name, gender, age, marital status,skills, country, city and address as keys for
student = {
    "first_name":None,
    "last_name":None,
    "gender":None,
    "age":None,
    "marital status":None,
    "skills":None,
    "country":None,
    "city":None,
    "address":None
}
#2d length of the student dictionary
print("\nThe length of Student dictionary is : ",len(student)) #use len()to find the length of the dictionary.
student = {
    "first_name":'Manasvi',
    "last_name":'Singam',
    "gender":'female',
    "age":'22',
    "marital status": 'Unmarried',
    "skills":['Pressure handling','Critical Thinking','Thinking out of box','logical Thinking'],
    "country":'United states',
    "city":'Overland park',
    "address":'5000w,cambridge square'
}
print("\nType :",type(student['skills']))#2e value of skills and check the data type
print("\nThe skills are:",student.get('skills'))#2f Modify the skills values by adding one or two skills
student['skills'].append("Analytical writing")
print("\nThe skills are adding an extra skill:",student.get('skills'))
#2g Get the dictionary keys as a list
print("\nThe keys in student dictionary :",student.keys())
#2h Get the dictionary values as a list
print("\nThe values in student dictionary :",student.values())
```

```
The length of Student dictionary is :  9

Type : <class 'list'>

The skills are: ['Pressure handling', 'Critical Thinking', 'Thinking out of box', 'logical Thinking']
```



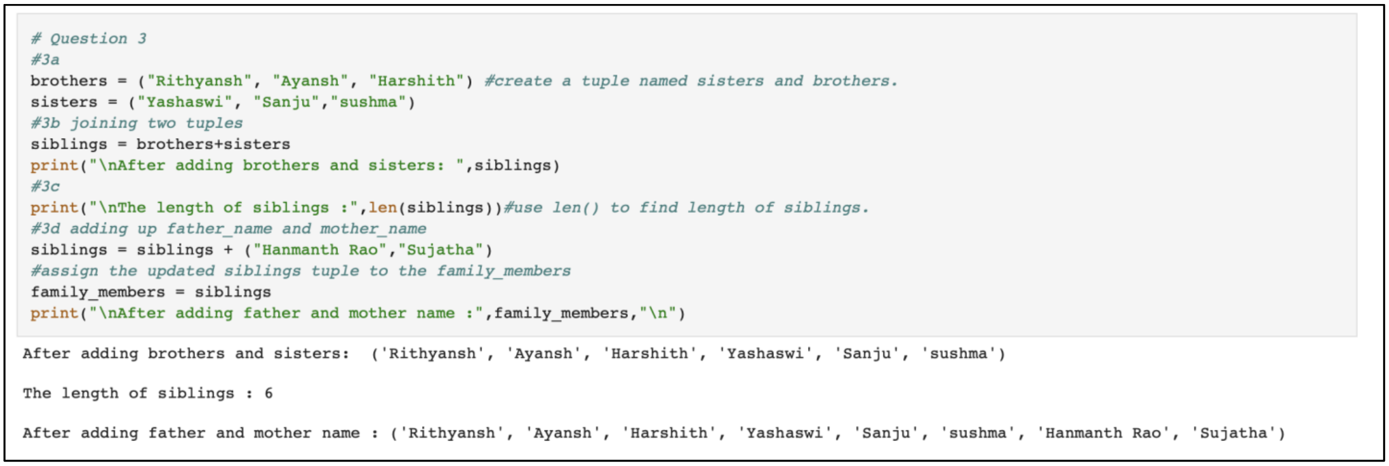
```
The skills are adding an extra skill: ['Pressure handling', 'Critical Thinking', 'Thinking out of box', 'logical Thinking', 'Analytical wr
iting']

The keys in student dictionary : dict_keys(['first_name', 'last_name', 'gender', 'age', 'marital status', 'skills', 'country', 'city', 'ad
dress'])

The values in student dictionary : dict_values(['Manasvi', 'Singam', 'female', '22', 'Unmarried', ['Pressure handling', 'Critical Thinking
', 'Thinking out of box', 'logical Thinking', 'Analytical writing'], 'United states', 'Overland park', '5000w,cambridge square'])
```

Explanation:

- Create an empty dictionary and Add name, color, breed, legs, age to the dog dictionary.

- Then create a student dictionary and add first name, last name, gender, age, marital status, skills, country, city and address as keys for the dictionary.
- Using len() get the length of the student dictionary.
- To check the data type use type ().
- Add new skills using append ().
- To get the keys use dictionary_name.keys() and to get values of the dictionary use dictionary_name.values().

3.



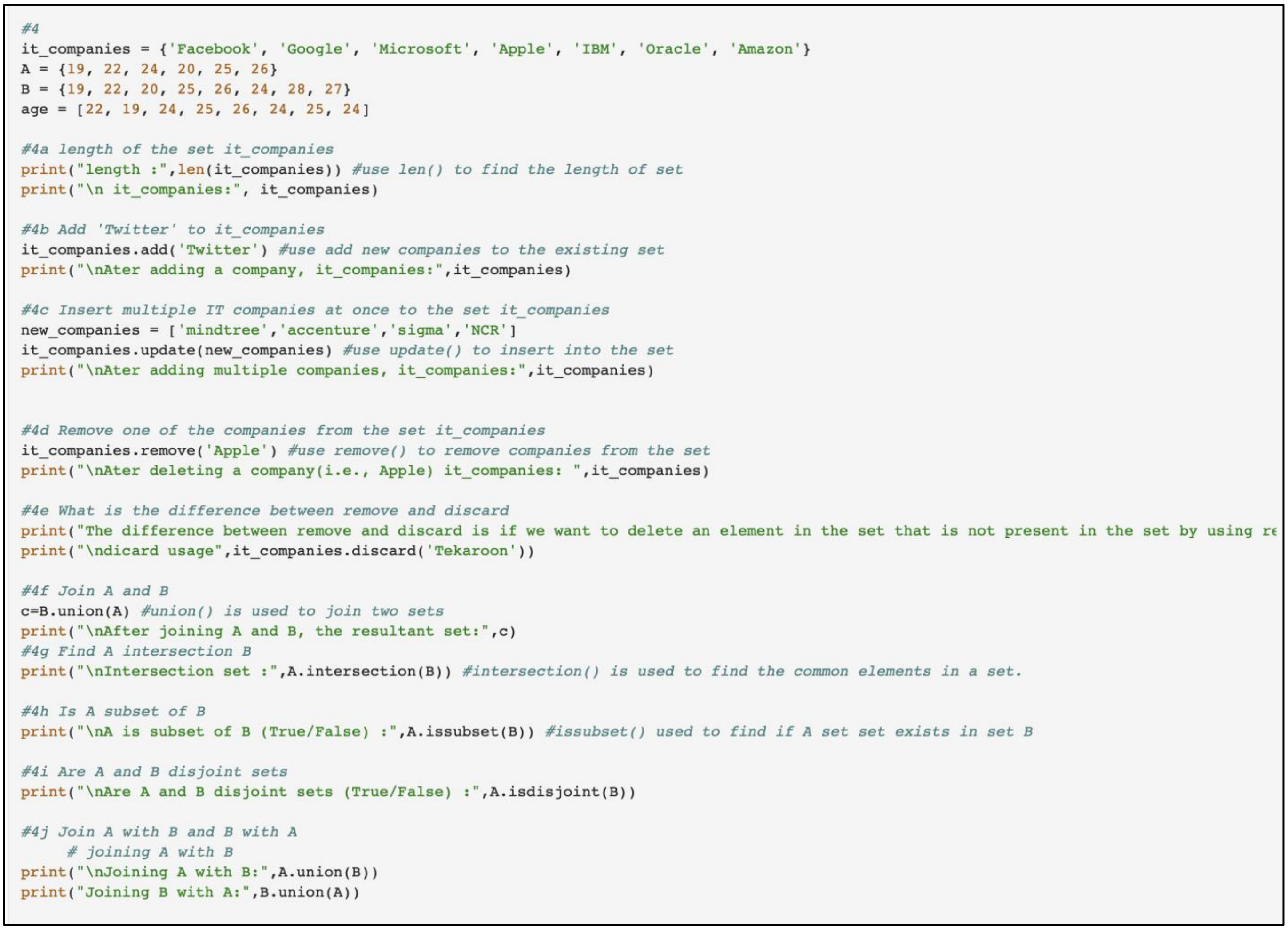
```python
# Question 3
#3a
brothers = ("Rithyansh", "Ayansh", "Harshith") #create a tuple named sisters and brothers.
sisters = ("Yashaswi", "Sanju","sushma")
#3b joining two tuples
siblings = brothers+sisters
print("\nAfter adding brothers and sisters: ",siblings)
#3c
print("\nThe length of siblings :",len(siblings))#use len() to find length of siblings.
#3d adding up father_name and mother_name
siblings = siblings + ("Hanmanth Rao","Sujatha")
#assign the updated siblings tuple to the family_members
family_members = siblings
print("\nAfter adding father and mother name :",family_members,"\n")
```

```
After adding brothers and sisters:  ('Rithyansh', 'Ayansh', 'Harshith', 'Yashaswi', 'Sanju', 'sushma')

The length of siblings : 6

After adding father and mother name : ('Rithyansh', 'Ayansh', 'Harshith', 'Yashaswi', 'Sanju', 'sushma', 'Hanmanth Rao', 'Sujatha')
```

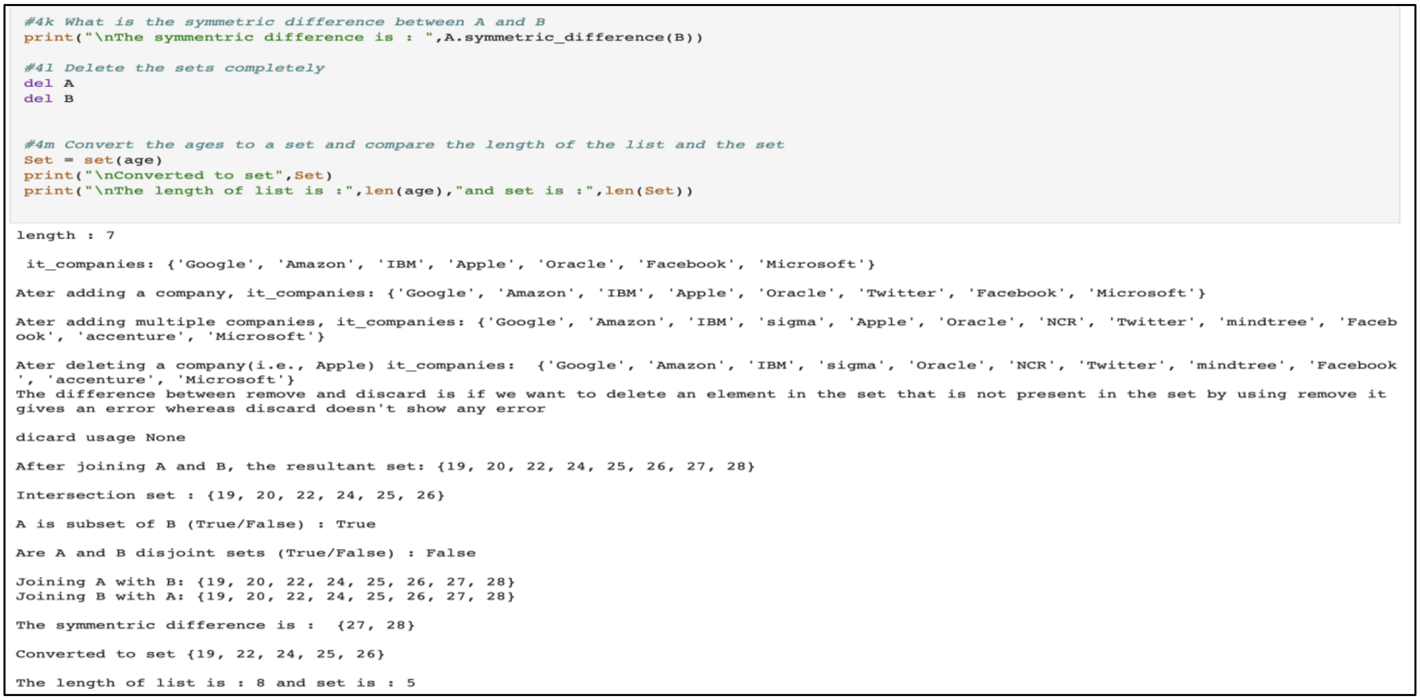
Explanation:

- Create a tuple containing the names of your sisters and your brothers.
- Add two tuples using “+”.
- Find the length of the tuple using len().
- Add the name of father and mother using “+”and assign it to family members.

4.



```
#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]

#4a length of the set it_companies
print("length :",len(it_companies)) #use len() to find the length of set
print("\n it_companies:", it_companies)

#4b Add 'Twitter' to it_companies
it_companies.add('Twitter') #use add new companies to the existing set
print("\nAter adding a company, it_companies:",it_companies)

#4c Insert multiple IT companies at once to the set it_companies
new_companies = ['mindtree','accenture','sigma','NCR']
it_companies.update(new_companies) #use update() to insert into the set
print("\nAter adding multiple companies, it_companies:",it_companies)


#4d Remove one of the companies from the set it_companies
it_companies.remove('Apple') #use remove() to remove companies from the set
print("\nAter deleting a company(i.e., Apple) it_companies: ",it_companies)

#4e What is the difference between remove and discard
print("The difference between remove and discard is if we want to delete an element in the set that is not present in the set by using re
print("\ndicard usage",it_companies.discard('Tekaroon'))

#4f Join A and B
c=B.union(A) #union() is used to join two sets
print("\nAfter joining A and B, the resultant set:",c)
#4g Find A intersection B
print("\nIntersection set :",A.intersection(B)) #intersection() is used to find the common elements in a set.

#4h Is A subset of B
print("\nA is subset of B (True/False) :",A.issubset(B)) #issubset() used to find if A set set exists in set B

#4i Are A and B disjoint sets
print("\nAre A and B disjoint sets (True/False) :",A.isdisjoint(B))

#4j Join A with B and B with A
     # joining A with B
print("\nJoining A with B:",A.union(B))
print("Joining B with A:",B.union(A))
```



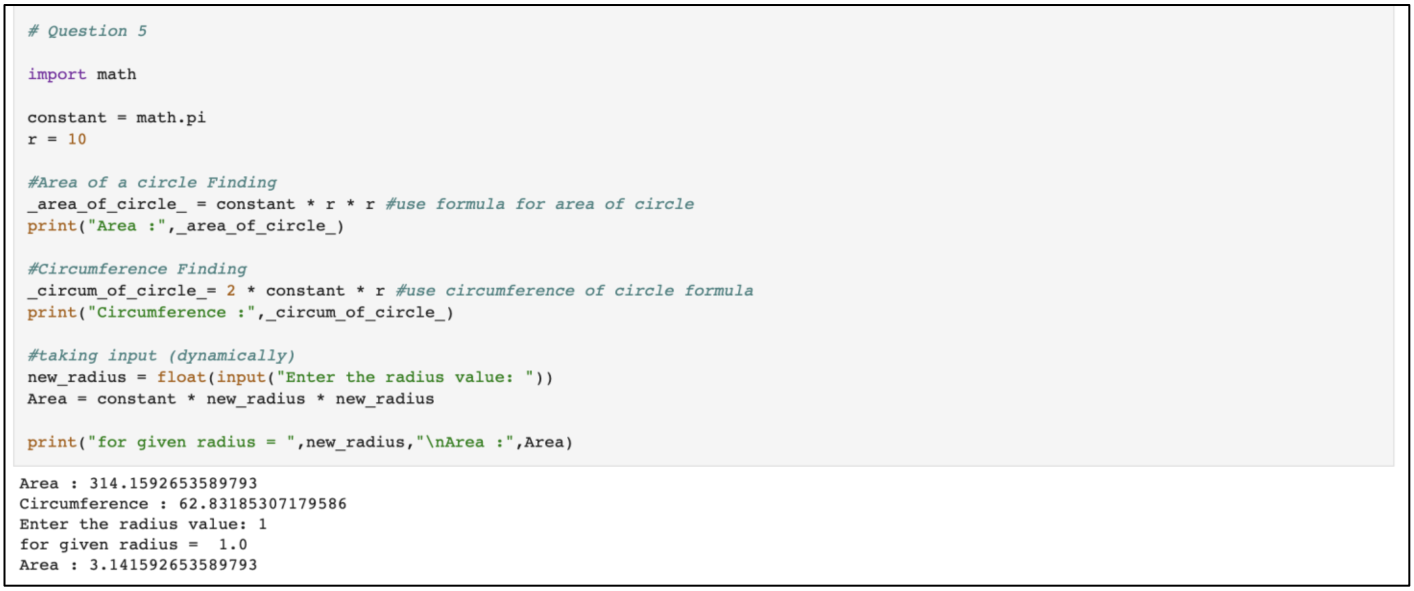
```
#4k What is the symmetric difference between A and B
print("\nThe symmentric difference is : ",A.symmetric_difference(B))

#4l Delete the sets completely
del A
del B


#4m Convert the ages to a set and compare the length of the list and the set
Set = set(age)
print("\nConverted to set",Set)
print("\nThe length of list is :",len(age),"and set is :",len(Set))
```

```
length : 7

 it_companies: {'Google', 'Amazon', 'IBM', 'Apple', 'Oracle', 'Facebook', 'Microsoft'}

Ater adding a company, it_companies: {'Google', 'Amazon', 'IBM', 'Apple', 'Oracle', 'Twitter', 'Facebook', 'Microsoft'}

Ater adding multiple companies, it_companies: {'Google', 'Amazon', 'IBM', 'sigma', 'Apple', 'Oracle', 'NCR', 'Twitter', 'mindtree', 'Faceb
ook', 'accenture', 'Microsoft'}

Ater deleting a company(i.e., Apple) it_companies:  {'Google', 'Amazon', 'IBM', 'sigma', 'Oracle', 'NCR', 'Twitter', 'mindtree', 'Facebook
', 'accenture', 'Microsoft'}
The difference between remove and discard is if we want to delete an element in the set that is not present in the set by using remove it
gives an error whereas discard doesn't show any error

dicard usage None

After joining A and B, the resultant set: {19, 20, 22, 24, 25, 26, 27, 28}

Intersection set : {19, 20, 22, 24, 25, 26}

A is subset of B (True/False) : True

Are A and B disjoint sets (True/False) : False

Joining A with B: {19, 20, 22, 24, 25, 26, 27, 28}
Joining B with A: {19, 20, 22, 24, 25, 26, 27, 28}

The symmentric difference is :  {27, 28}

Converted to set {19, 22, 24, 25, 26}

The length of list is : 8 and set is : 5
```

Explanation:

- Using len() find the length of the set it companies.
- Use add () to add the new company name into the set.
- Use update() to insert multiple companies at a time.
- Use remove(0 to remove companies from the set.
- Use union() to join two sets.
- Use intersection() to find the intersection between two sets.
- Use issubset() to find if one set exists in another.
- Use isdisjoint() to check if two sets are disjoint.
- Use del to completely delete the set.

5.



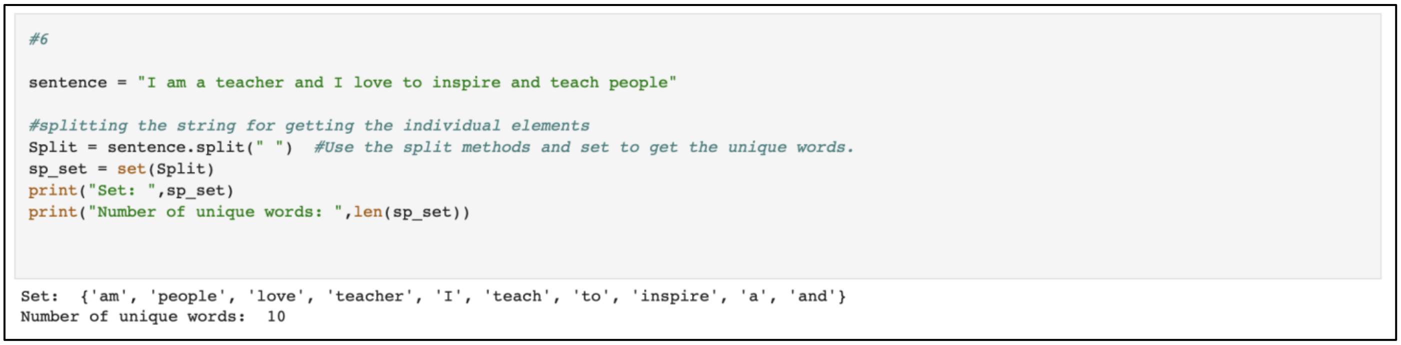
```
# Question 5

import math

constant = math.pi
r = 10

#Area of a circle Finding
_area_of_circle_ = constant * r * r #use formula for area of circle
print("Area :",_area_of_circle_)

#Circumference Finding
_circum_of_circle_= 2 * constant * r #use circumference of circle formula
print("Circumference :",_circum_of_circle_)

#taking input (dynamically)
new_radius = float(input("Enter the radius value: "))
Area = constant * new_radius * new_radius

print("for given radius = ",new_radius,"\nArea :",Area)
```

```
Area : 314.1592653589793
Circumference : 62.83185307179586
Enter the radius value: 1
for given radius =  1.0
Area : 3.141592653589793
```

Explanation:

- Use import math to extend mathematical functions.
- Math.pi gives pi value.
- Find the area of circle and circumference of circle.
- Take radius input dynamically from user and calculate the area of circle.

6.



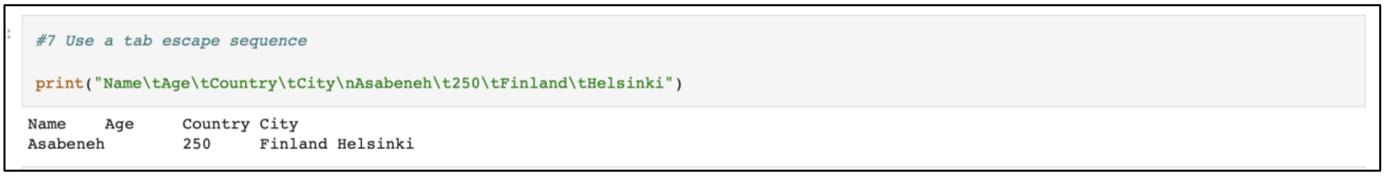
```
#6

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

#splitting the string for getting the individual elements
Split = sentence.split(" ")  #Use the split methods and set to get the unique words.
sp_set = set(Split)
print("Set: ",sp_set)
print("Number of unique words: ",len(sp_set))
```

```
Set:  {'am', 'people', 'love', 'teacher', 'I', 'teach', 'to', 'inspire', 'a', 'and'}
Number of unique words:  10
```

Explanation:

- Split the sentence using split () method.
- Convert the splitted string into set.
- Use len() to find the count of the unique elements.

7.



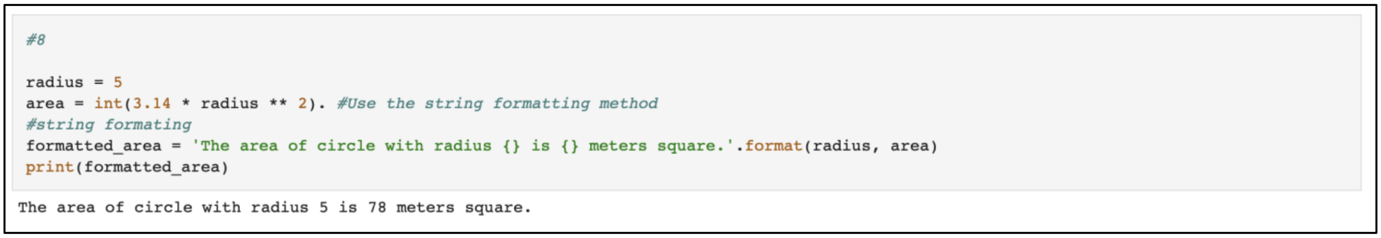
```
#7 Use a tab escape sequence

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

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

Explanation:

Use a tab escape sequence to get the desired output.

8.



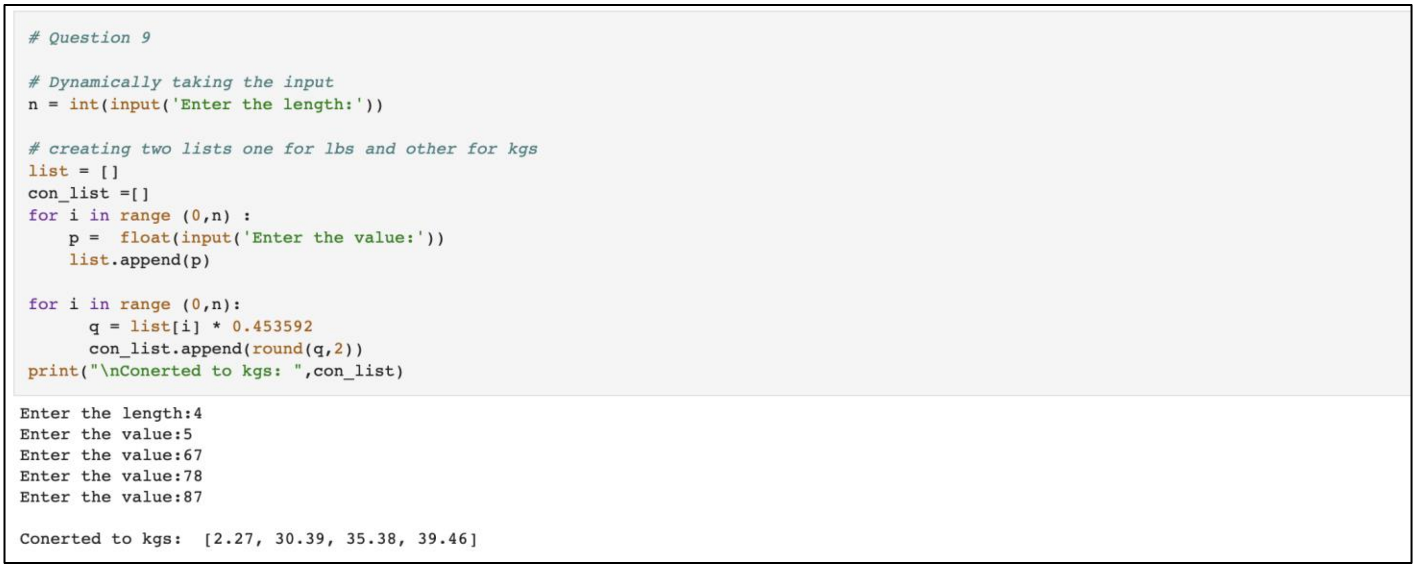
```
#8

radius = 5
area = int(3.14 * radius ** 2). #Use the string formatting method
#string formating
formatted_area = 'The area of circle with radius {} is {} meters square.'.format(radius, area)
print(formatted_area)
```

```
The area of circle with radius 5 is 78 meters square.
```

Explanation:

Use format () method to display the given output.

9.



```
# Question 9

# Dynamically taking the input
n = int(input('Enter the length:'))

# creating two lists one for lbs and other for kgs
list = []
con_list =[]
for i in range (0,n) :
    p =  float(input('Enter the value:'))
    list.append(p)

for i in range (0,n):
      q = list[i] * 0.453592
      con_list.append(round(q,2))
print("\nConerted to kgs: ",con_list)
```

```
Enter the length:4
Enter the value:5
Enter the value:67
Enter the value:78
Enter the value:87

Conerted to kgs:  [2.27, 30.39, 35.38, 39.46]
```

Explanation:

- Create two empty lists.
- convert the weights into kilograms in a separate list using for () loop.

10) $c_1 = 1, 2, 7, 10, 11$
$c_2 = 3, 6, 6$

Given the data should be divided into two parts equally.

FP (Training) = 1, 2, 3, 6
SP (Testing) = 6, 7, 10, 11

f	1	2	3	4	5	6	7	8	9	10	11	12	13
c_1	1	2					7			10	11		
c_2				3		6, 6							

$\underbrace{\qquad\qquad}_{\text{Training}} \qquad \underbrace{\qquad\qquad}_{\text{Testing}}$

~~Actual~~ Actual data

Training

c_1	c_2	c_2	c_2
1	2	3	6

Testing.

6	7	10	11
c_2	c_1	c_1	c_1

for Each training data
→ Measure distance for Every training point & find the K-closest points
→ Now get the most common class in those K predict that classes.

Now predicted data is 6 7 10 11
c_2 c_2 c_2 c_2

Euclidean distance, given $k = 3$
So, consider nearest 3 values

$\sqrt{(6-1)^2} = 5 \quad c_1$

$$\left.\begin{array}{l}\sqrt{(6-2)^2} = 4\\ \sqrt{(6-3)^2} = 3\\ \sqrt{(6-6)^2} = 0\end{array}\right\} C_2 \left(\text{nearest value}\right)$$

6 belongs to C_2

$\sqrt{(7-1)^2} = 6 \quad C_1$

$\sqrt{(7-2)^2} = 5 \quad C_1$

$\therefore$ 7 belongs to C_2

$\sqrt{(7-3)^2} = 4 \quad C_2$

$\sqrt{(7-6)^2} = 4 \quad C_2$

$\sqrt{(11-1)^2} = 10 \quad C_1$

$\therefore$ 11 belongs to C_2

$\sqrt{(11-2)^2} = 9 \quad C_1$

$\sqrt{(11-3)^2} = 8 \quad C_2$

$\sqrt{(11-6)^2} = 7 \quad C_2$

Actual Training set :	C_1	C_1	C_2	C_2
	1	2	3	6

Actual Testing :	6	7	10	11
	C_2	C_1	C_1	C_1

Predicted Testing Outputs

6	7	10	11
C_2	C_2	C_2	C_2

Confusion Matrix prediction

		c_1	c_2
Truth	c_1	TN	FP
	c_2	TN	FP

Actual

5	7	10	11
c_2	c_1	c_1	c_1

predicted

6	7	10	11
c_2	c_2	c_2	c_2

$$\text{Accuracy} = \frac{(TP+TN)}{P+N} = \frac{1+0}{4}$$

$$= 0.25$$

$$\text{Senstivity} = \frac{TP}{TP+FN} = \frac{TP}{P} = \frac{1}{1+0} = 1.$$

$$\text{Specificity} = \frac{TN}{FP+TN} = \frac{TN}{N} = \frac{0}{(3+0)} = 0. //$$