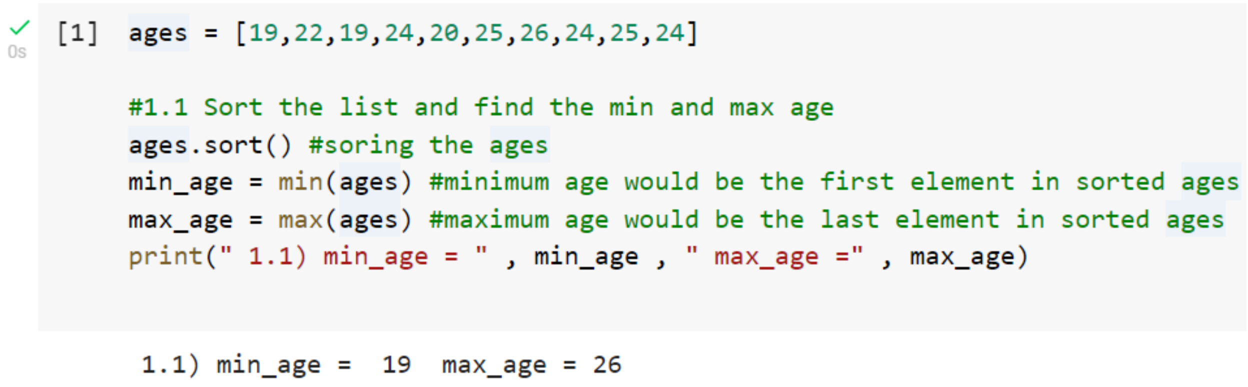
Assignment - 1
Machine Learning

Divya Sai Ajay Jasti
700741296

Question 1

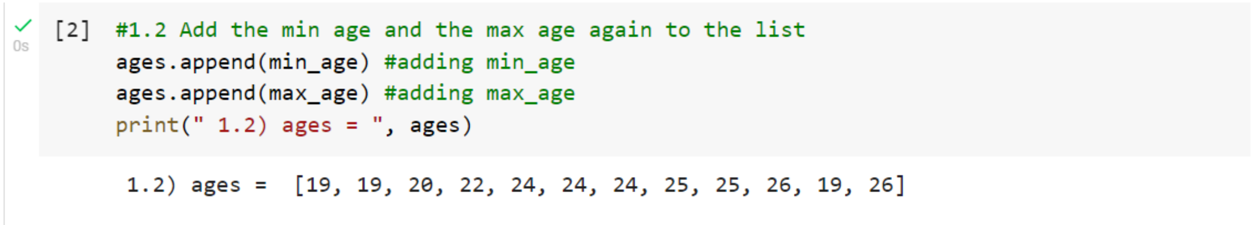
The following is a list of 10 students ages:
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]

• Sort the list and find the min and max age



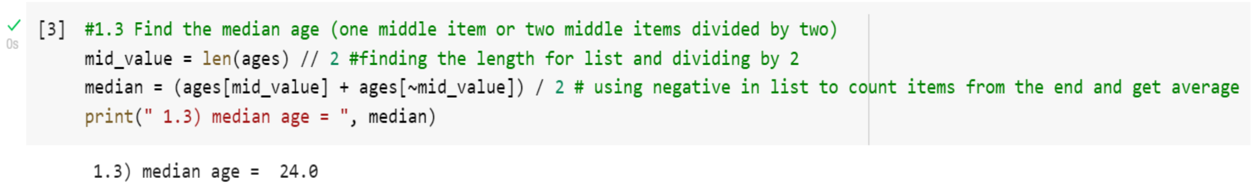
```
[1] ages = [19,22,19,24,20,25,26,24,25,24]

    #1.1 Sort the list and find the min and max age
    ages.sort() #soring the ages
    min_age = min(ages) #minimum age would be the first element in sorted ages
    max_age = max(ages) #maximum age would be the last element in sorted ages
    print(" 1.1) min_age = " , min_age , " max_age =" , max_age)

     1.1) min_age =  19  max_age = 26
```

• Add the min age and the max age again to the list



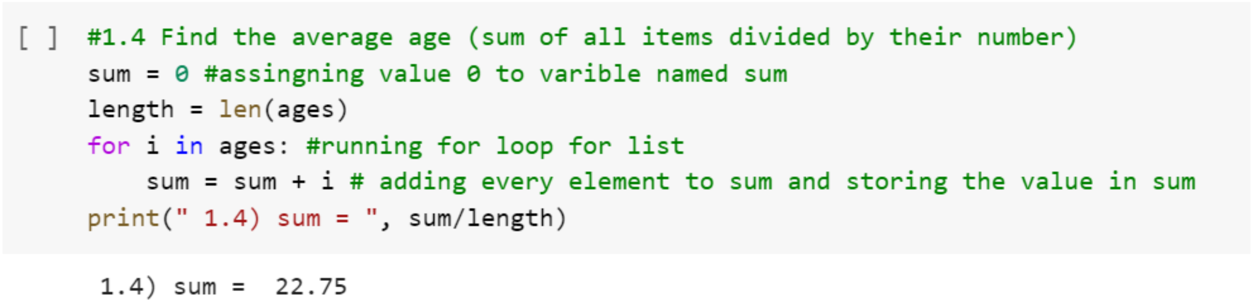
```
[2] #1.2 Add the min age and the max age again to the list
    ages.append(min_age) #adding min_age
    ages.append(max_age) #adding max_age
    print(" 1.2) ages = ", ages)

     1.2) ages =  [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
```

• Find the median age (one middle item or two middle items divided by two)



```
[3] #1.3 Find the median age (one middle item or two middle items divided by two)
    mid_value = len(ages) // 2 #finding the length for list and dividing by 2
    median = (ages[mid_value] + ages[~mid_value]) / 2 # using negative in list to count items from the end and get average
    print(" 1.3) median age = ", median)

     1.3) median age =  24.0
```

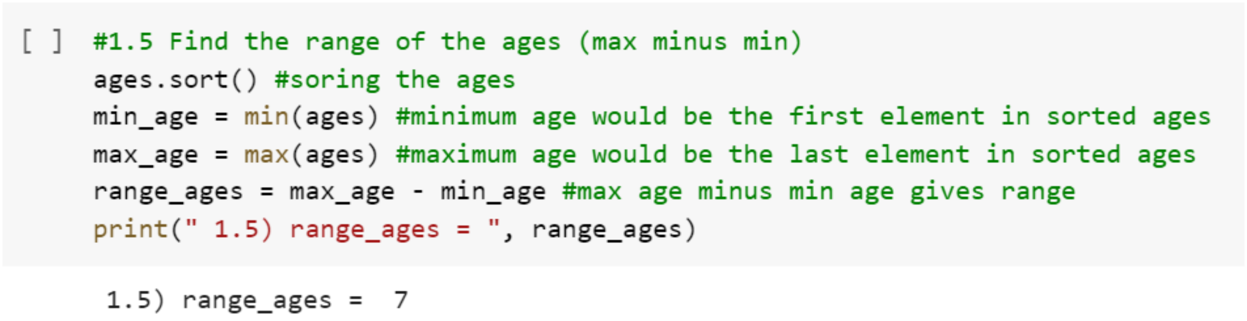
- **Find the average age (sum of all items divided by their number)**



```
[ ]  #1.4 Find the average age (sum of all items divided by their number)
     sum = 0 #assingning value 0 to varible named sum
     length = len(ages)
     for i in ages: #running for loop for list
         sum = sum + i # adding every element to sum and storing the value in sum
     print(" 1.4) sum = ", sum/length)
```

```
 1.4) sum =  22.75
```

- **Find the range of the ages (max minus min)**

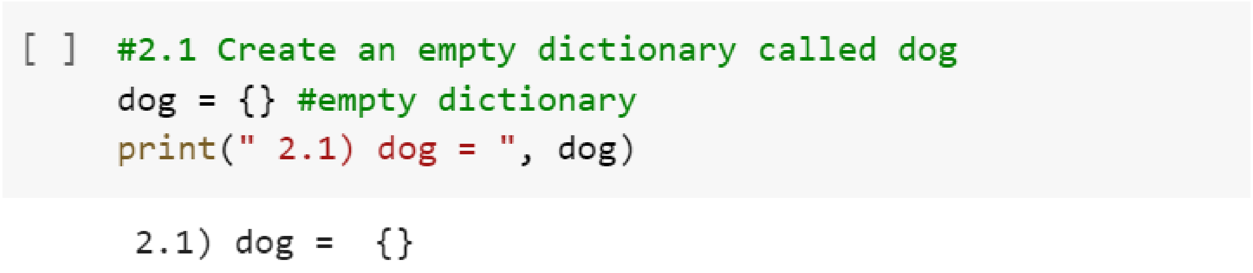


```
[ ]  #1.5 Find the range of the ages (max minus min)
     ages.sort() #soring the ages
     min_age = min(ages) #minimum age would be the first element in sorted ages
     max_age = max(ages) #maximum age would be the last element in sorted ages
     range_ages = max_age - min_age #max age minus min age gives range
     print(" 1.5) range_ages = ", range_ages)
```

```
 1.5) range_ages =  7
```

Question 2

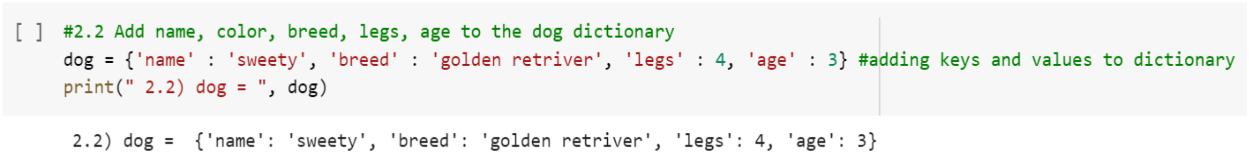
- **Create an empty dictionary called dog**



```
[ ]  #2.1 Create an empty dictionary called dog
     dog = {} #empty dictionary
     print(" 2.1) dog = ", dog)
```

```
 2.1) dog =  {}
```

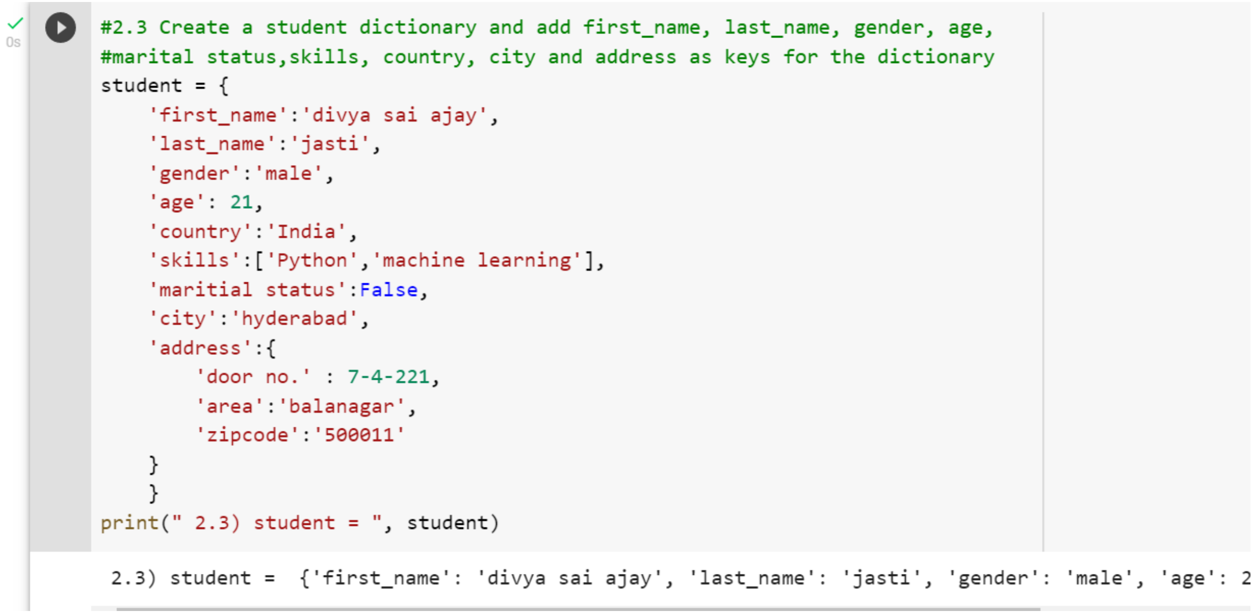
- **Add name, color, breed, legs, age to the dog dictionary**



```
[ ]  #2.2 Add name, color, breed, legs, age to the dog dictionary
     dog = {'name' : 'sweety', 'breed' : 'golden retriver', 'legs' : 4, 'age' : 3} #adding keys and values to dictionary
     print(" 2.2) dog = ", dog)
```

```
 2.2) dog =  {'name': 'sweety', 'breed': 'golden retriver', 'legs': 4, 'age': 3}
```

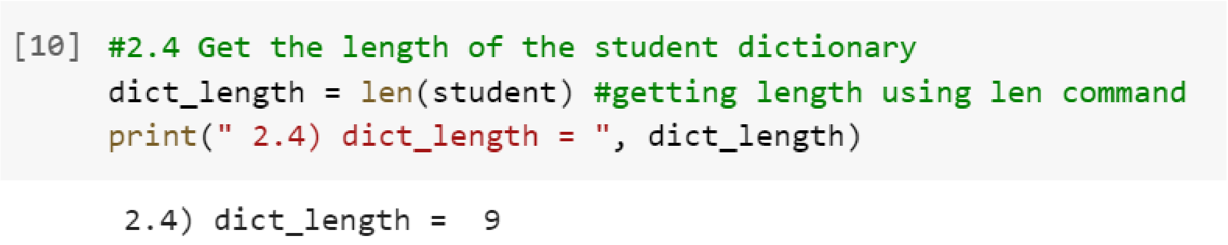
- **Create a student dictionary and add first_name, last_name, gender, age, marital status, skills, country, city and address as keys for the dictionary**



```
#2.3 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':'divya sai ajay',
    'last_name':'jasti',
    'gender':'male',
    'age': 21,
    'country':'India',
    'skills':['Python','machine learning'],
    'maritial status':False,
    'city':'hyderabad',
    'address':{
        'door no.' : 7-4-221,
        'area':'balanagar',
        'zipcode':'500011'
    }
    }
print(" 2.3) student = ", student)
```

```
2.3) student =  {'first_name': 'divya sai ajay', 'last_name': 'jasti', 'gender': 'male', 'age': 2
```

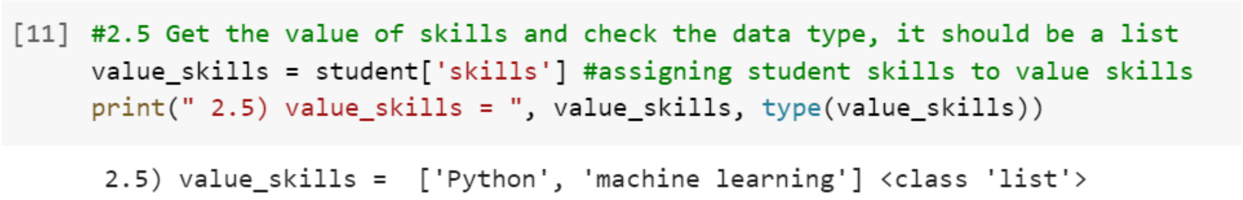
- **Get the length of the student dictionary**



```
#2.4 Get the length of the student dictionary
dict_length = len(student) #getting length using len command
print(" 2.4) dict_length = ", dict_length)
```

```
2.4) dict_length =  9
```

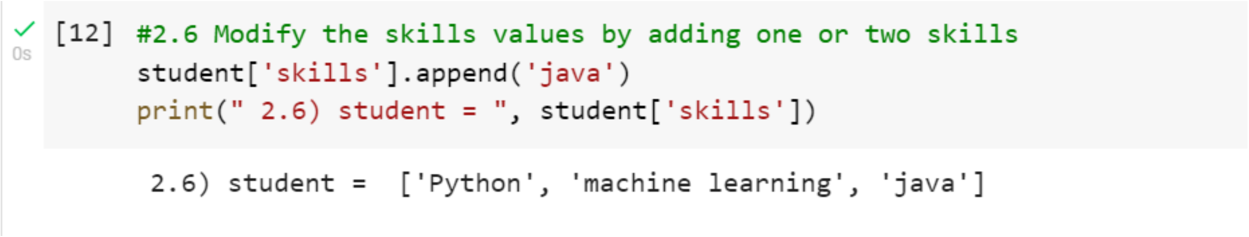
- **Get the value of skills and check the data type, it should be a list**



```
#2.5 Get the value of skills and check the data type, it should be a list
value_skills = student['skills'] #assigning student skills to value skills
print(" 2.5) value_skills = ", value_skills, type(value_skills))
```

```
2.5) value_skills =  ['Python', 'machine learning'] <class 'list'>
```

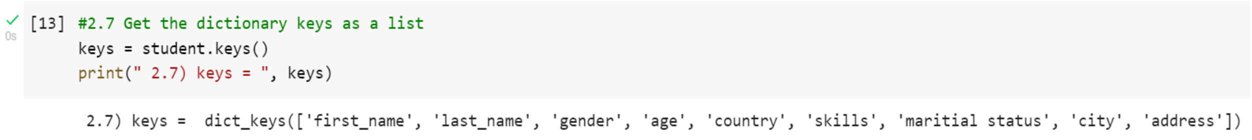
- **Modify the skills values by adding one or two skills**



```
#2.6 Modify the skills values by adding one or two skills
student['skills'].append('java')
print(" 2.6) student = ", student['skills'])
```

```
2.6) student =  ['Python', 'machine learning', 'java']
```

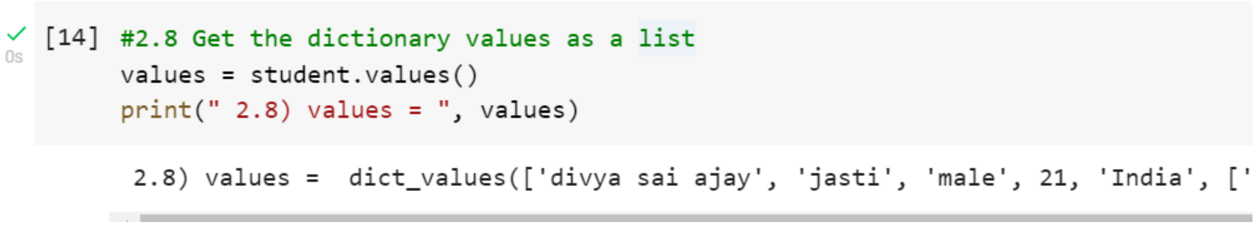
- **Get the dictionary keys as a list**



```
[13] #2.7 Get the dictionary keys as a list
     keys = student.keys()
     print(" 2.7) keys = ", keys)
```

```
 2.7) keys =  dict_keys(['first_name', 'last_name', 'gender', 'age', 'country', 'skills', 'maritial status', 'city', 'address'])
```

- **Get the dictionary values as a list**

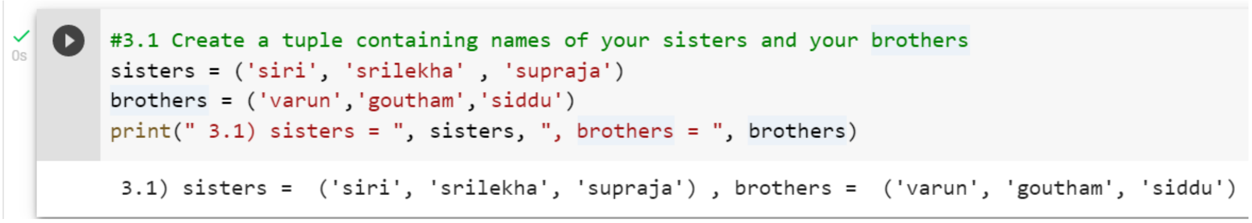


```
[14] #2.8 Get the dictionary values as a list
     values = student.values()
     print(" 2.8) values = ", values)
```

```
 2.8) values =  dict_values(['divya sai ajay', 'jasti', 'male', 21, 'India', ['
```

Question 3

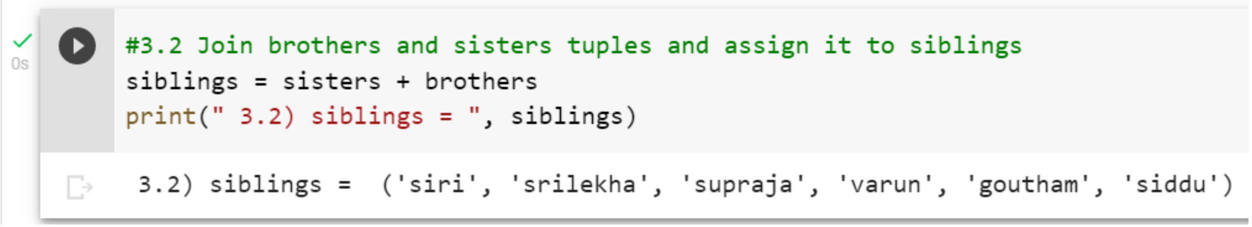
- **Create a tuple containing names of your sisters and your brothers (imaginary siblings are fine)**



```
#3.1 Create a tuple containing names of your sisters and your brothers
sisters = ('siri', 'srilekha' , 'supraja')
brothers = ('varun','goutham','siddu')
print(" 3.1) sisters = ", sisters, ", brothers = ", brothers)
```

```
 3.1) sisters =  ('siri', 'srilekha', 'supraja') , brothers =  ('varun', 'goutham', 'siddu')
```

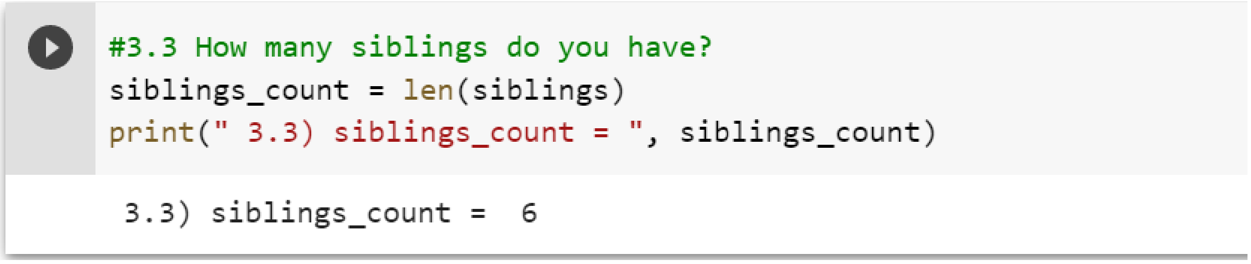
- **Join brothers and sisters tuples and assign it to siblings**



```
#3.2 Join brothers and sisters tuples and assign it to siblings
siblings = sisters + brothers
print(" 3.2) siblings = ", siblings)
```

```
 3.2) siblings =  ('siri', 'srilekha', 'supraja', 'varun', 'goutham', 'siddu')
```

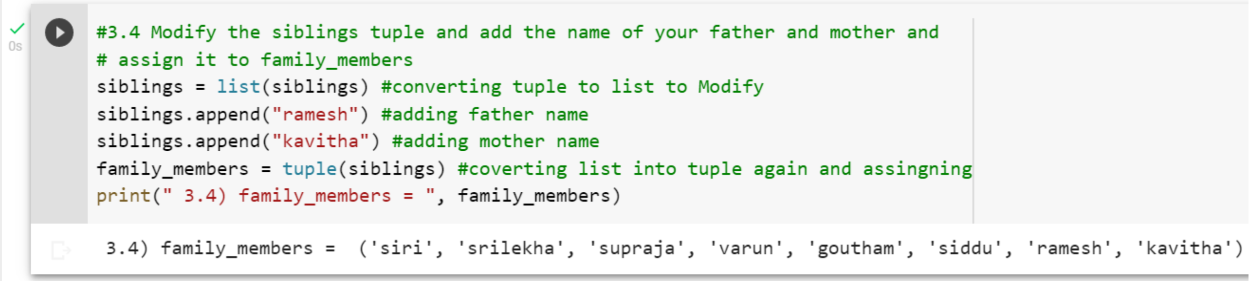
- **How many siblings do you have?**



```
#3.3 How many siblings do you have?
siblings_count = len(siblings)
print(" 3.3) siblings_count = ", siblings_count)
```

```
 3.3) siblings_count =  6
```

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



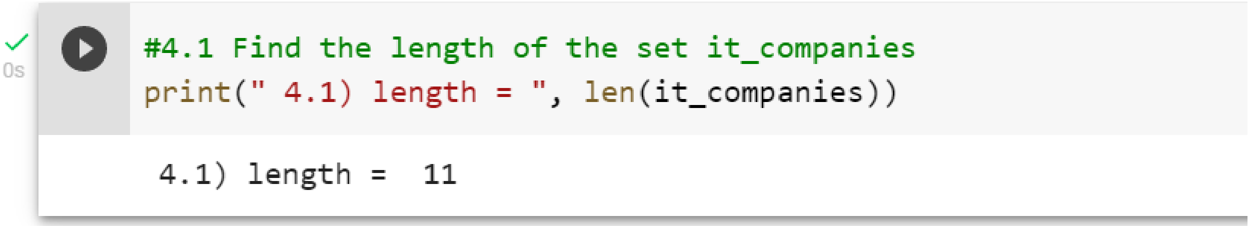
```
#3.4 Modify the siblings tuple and add the name of your father and mother and
# assign it to family_members
siblings = list(siblings) #converting tuple to list to Modify
siblings.append("ramesh") #adding father name
siblings.append("kavitha") #adding mother name
family_members = tuple(siblings) #coverting list into tuple again and assingning
print(" 3.4) family_members = ", family_members)
```

```
3.4) family_members =  ('siri', 'srilekha', 'supraja', 'varun', 'goutham', 'siddu', 'ramesh', 'kavitha')
```

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**

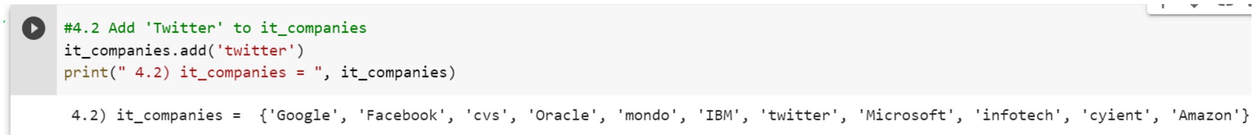


```
#4.1 Find the length of the set it_companies
print(" 4.1) length = ", len(it_companies))
```

```
4.1) length =  11
```

- **Add 'Twitter' to it_companies**

Adding twitter using **add** keyword to it_companies set.

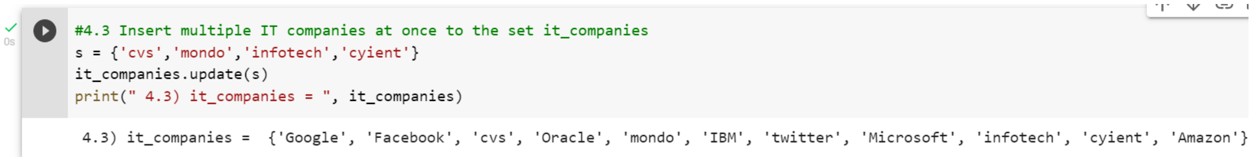


```
#4.2 Add 'Twitter' to it_companies
it_companies.add('twitter')
print(" 4.2) it_companies = ", it_companies)
```

```
4.2) it_companies =  {'Google', 'Facebook', 'cvs', 'Oracle', 'mondo', 'IBM', 'twitter', 'Microsoft', 'infotech', 'cyient', 'Amazon'}
```

- **Insert multiple IT companies at once to the set it_companies**

Inserting multiple values by using **update** keyword

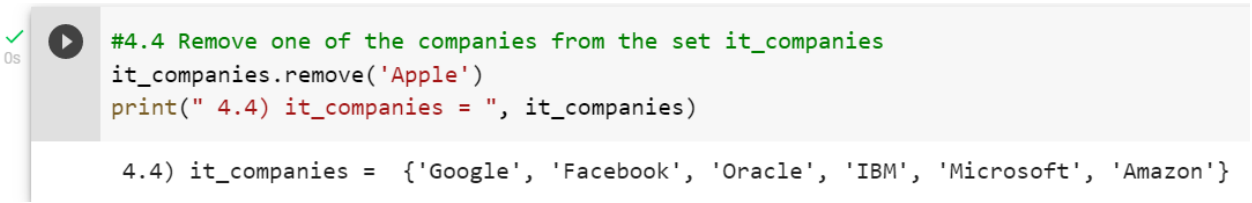


```
#4.3 Insert multiple IT companies at once to the set it_companies
s = {'cvs','mondo','infotech','cyient'}
it_companies.update(s)
print(" 4.3) it_companies = ", it_companies)
```

```
4.3) it_companies =  {'Google', 'Facebook', 'cvs', 'Oracle', 'mondo', 'IBM', 'twitter', 'Microsoft', 'infotech', 'cyient', 'Amazon'}
```

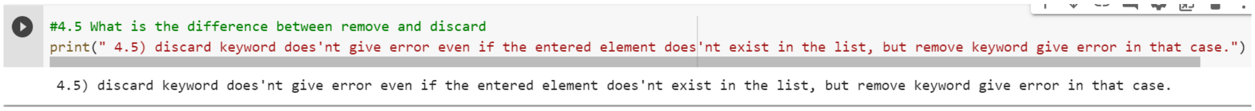
- **Remove one of the companies from the set it_companies**

Removing element by using **remove** keyword



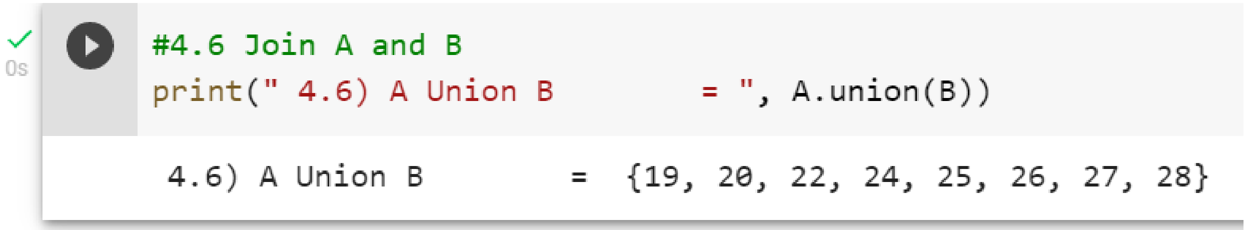
```
#4.4 Remove one of the companies from the set it_companies
it_companies.remove('Apple')
print(" 4.4) it_companies = ", it_companies)

 4.4) it_companies =  {'Google', 'Facebook', 'Oracle', 'IBM', 'Microsoft', 'Amazon'}
```

- **What is the difference between remove and discard**



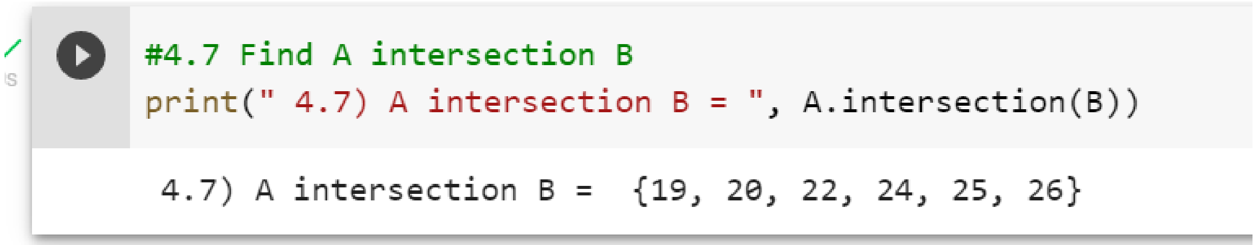
```
#4.5 What is the difference between remove and discard
print(" 4.5) discard keyword does'nt give error even if the entered element does'nt exist in the list, but remove keyword give error in that case.")

 4.5) discard keyword does'nt give error even if the entered element does'nt exist in the list, but remove keyword give error in that case.
```

- **Join A and B**



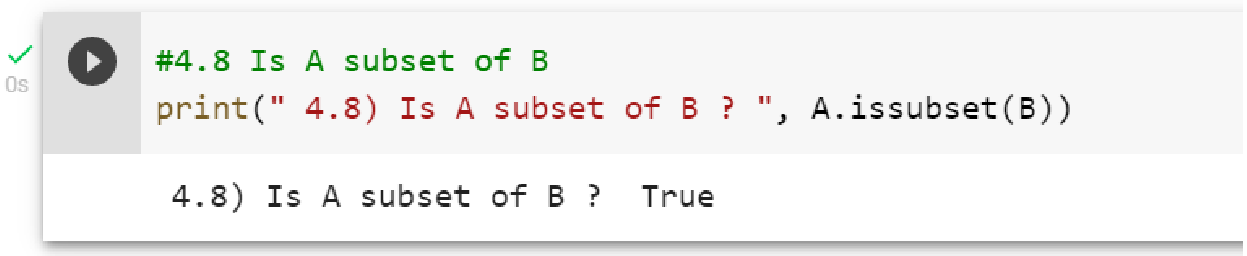
```
#4.6 Join A and B
print(" 4.6) A Union B          = ", A.union(B))

 4.6) A Union B          =  {19, 20, 22, 24, 25, 26, 27, 28}
```

- **Find A intersection B**



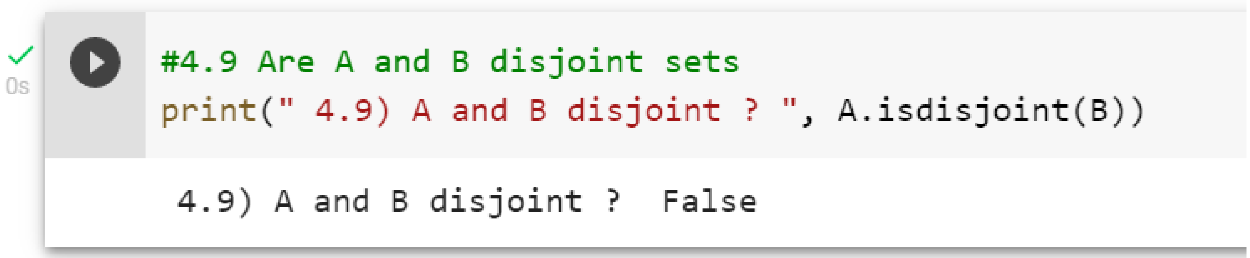
```
#4.7 Find A intersection B
print(" 4.7) A intersection B = ", A.intersection(B))

 4.7) A intersection B =  {19, 20, 22, 24, 25, 26}
```

- **Is A subset of B**



```
#4.8 Is A subset of B
print(" 4.8) Is A subset of B ? ", A.issubset(B))

 4.8) Is A subset of B ?  True
```

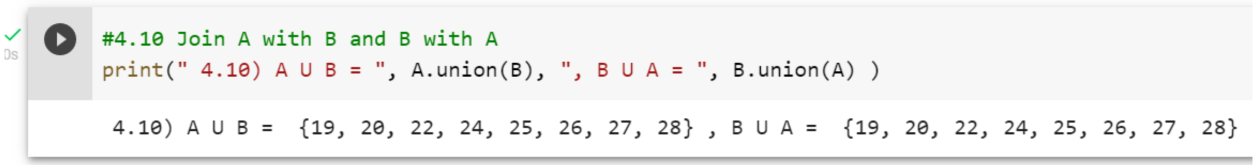
- **Are A and B disjoint sets**



```
#4.9 Are A and B disjoint sets
print(" 4.9) A and B disjoint ? ", A.isdisjoint(B))
```

```
4.9) A and B disjoint ?  False
```

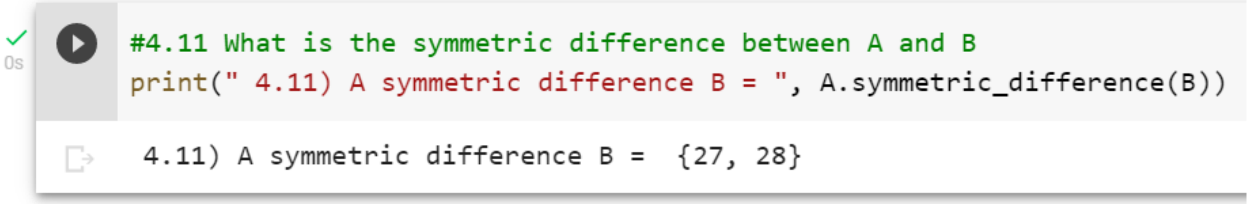
- **Join A with B and B with A**



```
#4.10 Join A with B and B with A
print(" 4.10) A U B = ", A.union(B), ", B U A = ", B.union(A) )
```

```
4.10) A U B =  {19, 20, 22, 24, 25, 26, 27, 28} , B U A =  {19, 20, 22, 24, 25, 26, 27, 28}
```

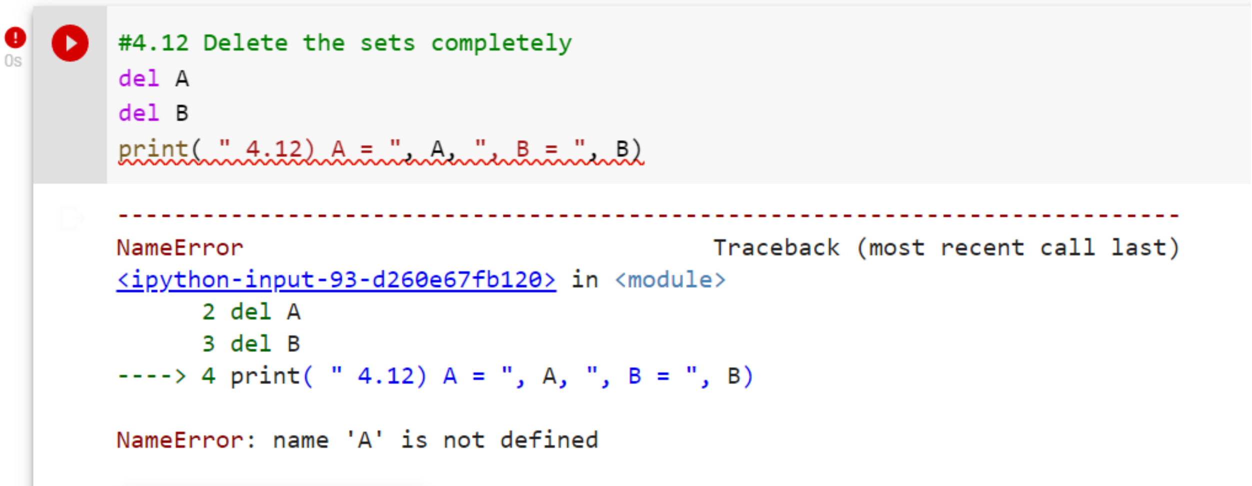
- **What is the symmetric difference between A and B**



```
#4.11 What is the symmetric difference between A and B
print(" 4.11) A symmetric difference B = ", A.symmetric_difference(B))
```

```
4.11) A symmetric difference B =  {27, 28}
```

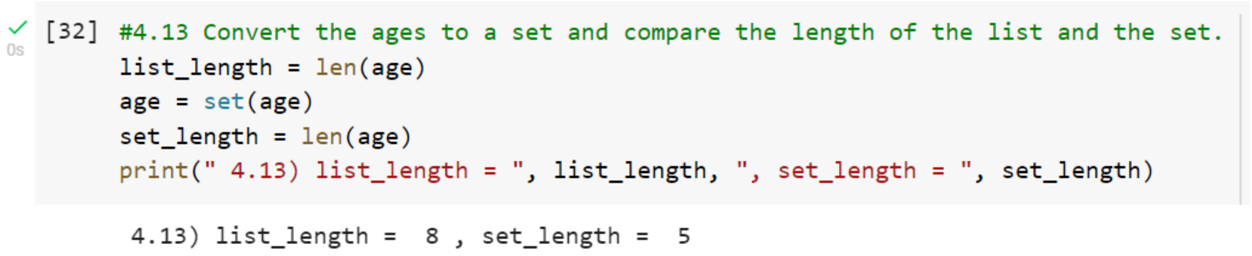
- **Delete the sets completely**



```
#4.12 Delete the sets completely
del A
del B
print( " 4.12) A = ", A, ", B = ", B)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-93-d260e67fb120> in <module>
      2 del A
      3 del B
----> 4 print( " 4.12) A = ", A, ", B = ", B)

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

- **Convert the ages to a set and compare the length of the list and the set.**



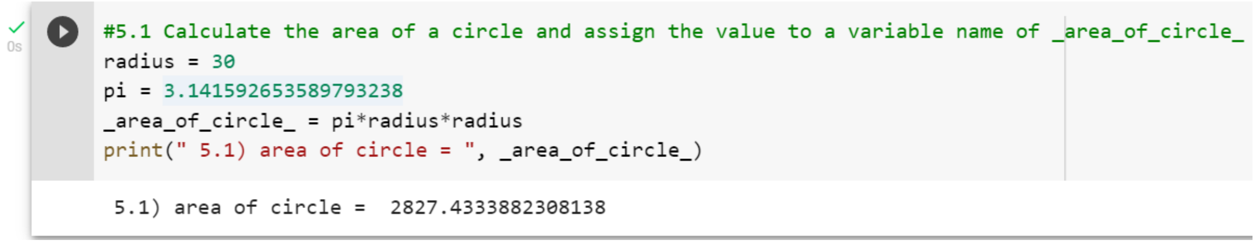
```
#4.13 Convert the ages to a set and compare the length of the list and the set.
list_length = len(age)
age = set(age)
set_length = len(age)
print(" 4.13) list_length = ", list_length, ", set_length = ", set_length)
```

```
 4.13) list_length =  8 , set_length =  5
```

Question 5

The radius of a circle is 30 meters.

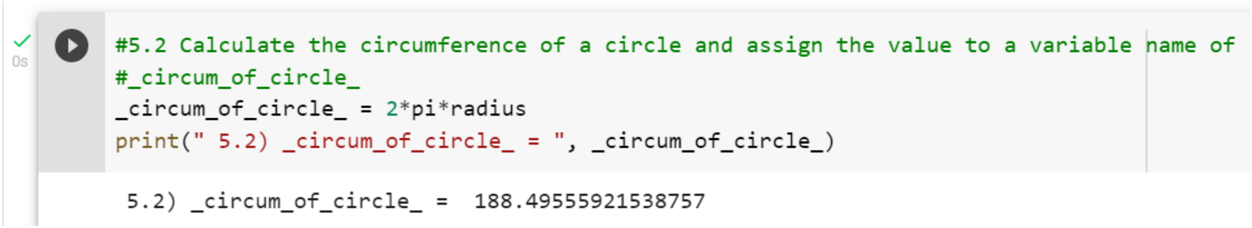
- **Calculate the area of a circle and assign the value to a variable name of _area_of_circle_**



```
#5.1 Calculate the area of a circle and assign the value to a variable name of _area_of_circle_
radius = 30
pi = 3.141592653589793238
_area_of_circle_ = pi*radius*radius
print(" 5.1) area of circle = ", _area_of_circle_)
```

```
 5.1) area of circle =  2827.4333882308138
```

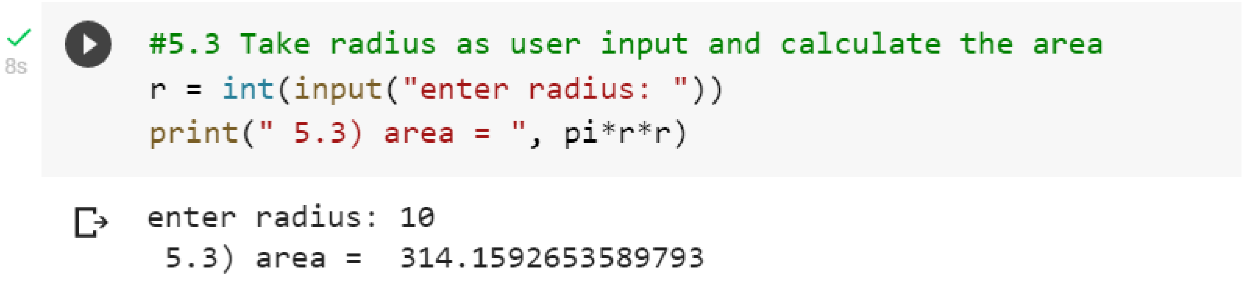
- **Calculate the circumference of a circle and assign the value to a variable name of _circum_of_circle_**



```
#5.2 Calculate the circumference of a circle and assign the value to a variable name of
#_circum_of_circle_
_circum_of_circle_ = 2*pi*radius
print(" 5.2) _circum_of_circle_ = ", _circum_of_circle_)
```

```
 5.2) _circum_of_circle_ =  188.49555921538757
```

- **Take radius as user input and calculate the area.**



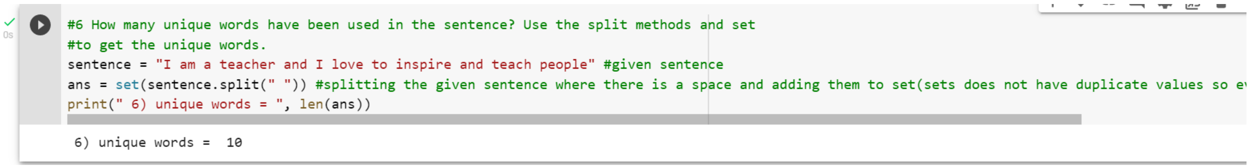
```
#5.3 Take radius as user input and calculate the area
r = int(input("enter radius: "))
print(" 5.3) area = ", pi*r*r)
```

```
enter radius: 10
 5.3) area =  314.1592653589793
```

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.

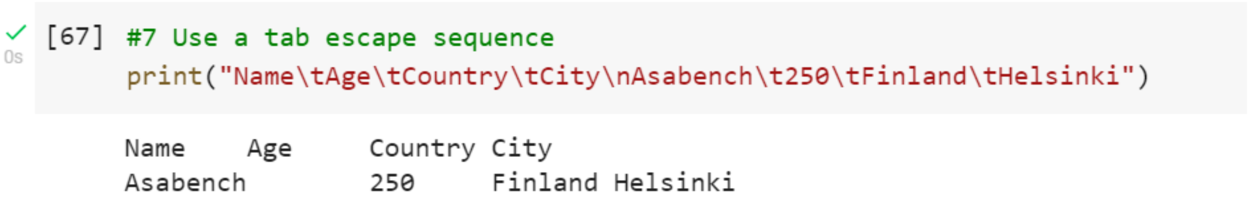


```
#6 How many unique words have been used in the sentence? Use the split methods and set
#to get the unique words.
sentence = "I am a teacher and I love to inspire and teach people" #given sentence
ans = set(sentence.split(" ")) #splitting the given sentence where there is a space and adding them to set(sets does not have duplicate values so e
print(" 6) unique words = ", len(ans))
```

```
 6) unique words =  10
```

Question 7

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

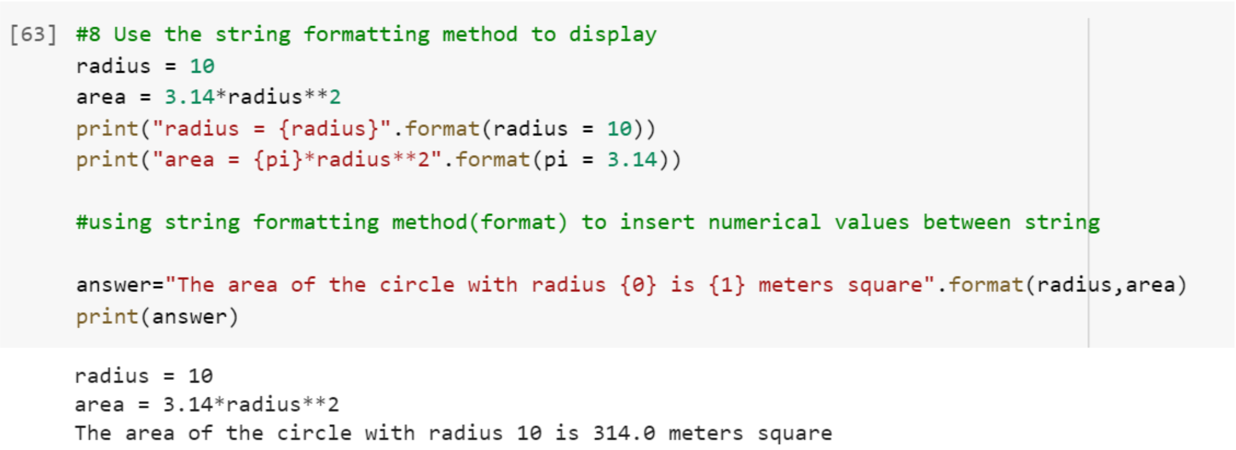


```
[67] #7 Use a tab escape sequence
     print("Name\tAge\tCountry\tCity\nAsabench\t250\tFinland\tHelsinki")
```

```
Name    Age     Country City
Asabench        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.”

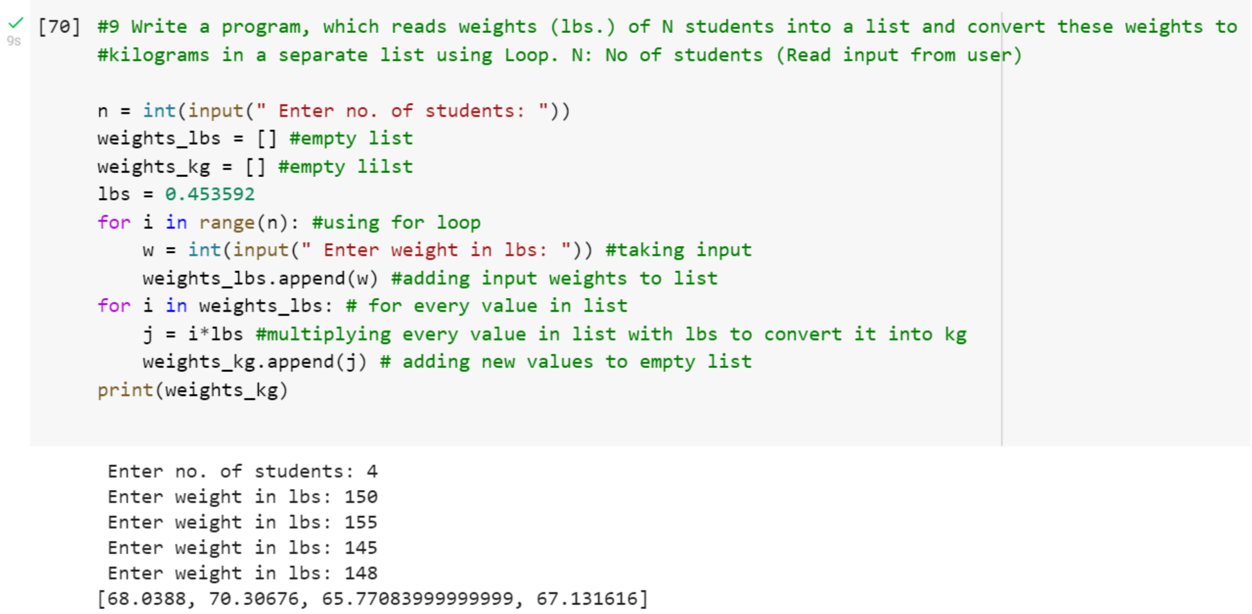


```
[63] #8 Use the string formatting method to display
     radius = 10
     area = 3.14*radius**2
     print("radius = {radius}".format(radius = 10))
     print("area = {pi}*radius**2".format(pi = 3.14))

     #using string formatting method(format) to insert numerical values between string

     answer="The area of the circle with radius {0} is {1} meters square".format(radius,area)
     print(answer)
```

```
radius = 10
area = 3.14*radius**2
The area of the circle with radius 10 is 314.0 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
Ex: L1: [150, 155, 145, 148]
Output: [68.03, 70.3, 65.77, 67.13]

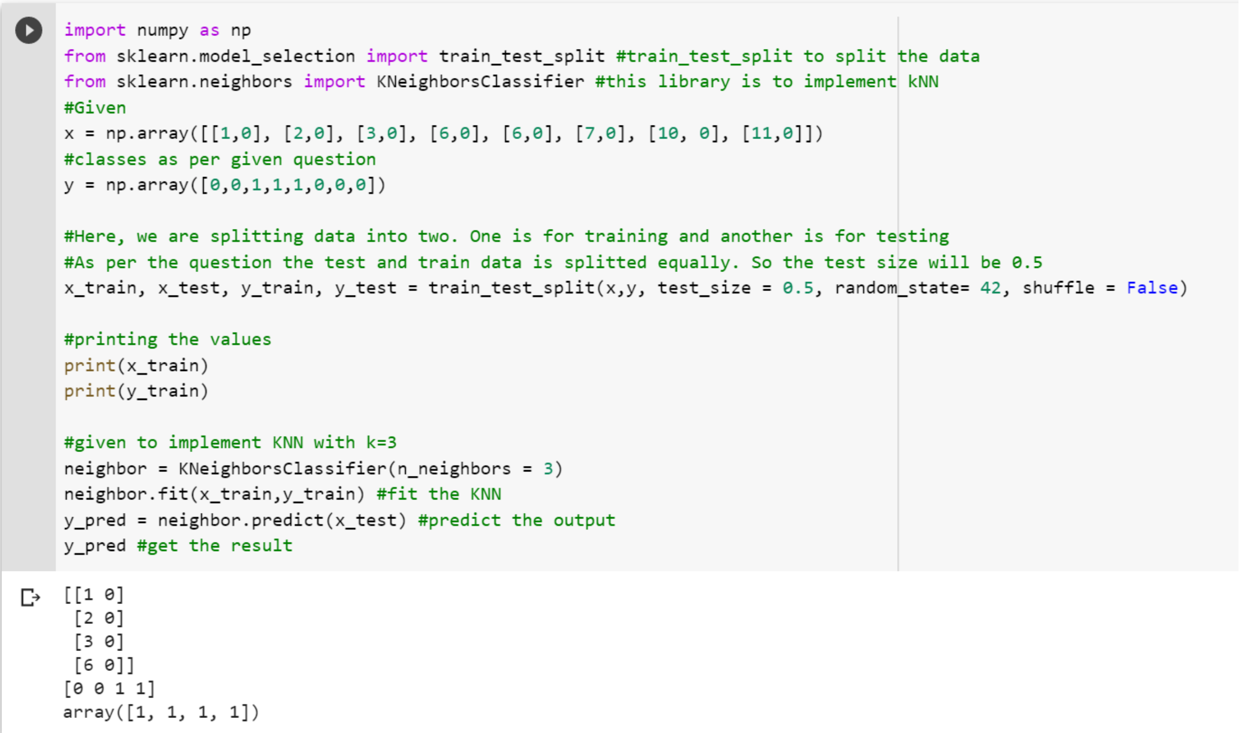


```
#9 Write a program, which reads weights (lbs.) of N students into a list and convert these weights to
#kilograms in a separate list using Loop. N: No of students (Read input from user)

n = int(input(" Enter no. of students: "))
weights_lbs = [] #empty list
weights_kg = [] #empty lilst
lbs = 0.453592
for i in range(n): #using for loop
    w = int(input(" Enter weight in lbs: ")) #taking input
    weights_lbs.append(w) #adding input weights to list
for i in weights_lbs: # for every value in list
    j = i*lbs #multiplying every value in list with lbs to convert it into kg
    weights_kg.append(j) # adding new values to empty list
print(weights_kg)
```

```
 Enter no. of students: 4
 Enter weight in lbs: 150
 Enter weight in lbs: 155
 Enter weight in lbs: 145
 Enter weight in lbs: 148
[68.0388, 70.30676, 65.77083999999999, 67.131616]
```

Question 10

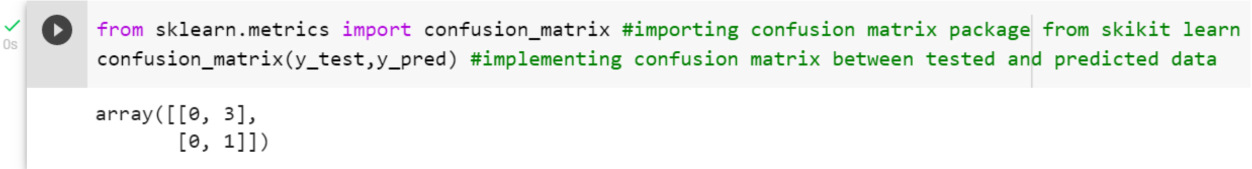
Divide this data equally into two parts. Use first part as training and second part as testing. Using KNN classifier, for K=3, what would be the predicted outputs for the test samples? Show how you arrived at your answer.



```
import numpy as np
from sklearn.model_selection import train_test_split #train_test_split to split the data
from sklearn.neighbors import KNeighborsClassifier #this library is to implement kNN
#Given
x = np.array([[1,0], [2,0], [3,0], [6,0], [6,0], [7,0], [10, 0], [11,0]])
#classes as per given question
y = np.array([0,0,1,1,1,0,0,0])

#Here, we are splitting data into two. One is for training and another is for testing
#As per the question the test and train data is splitted equally. So the test size will be 0.5
x_train, x_test, y_train, y_test = train_test_split(x,y, test_size = 0.5, random_state= 42, shuffle = False)

#printing the values
print(x_train)
print(y_train)

#given to implement KNN with k=3
neighbor = KNeighborsClassifier(n_neighbors = 3)
neighbor.fit(x_train,y_train) #fit the KNN
y_pred = neighbor.predict(x_test) #predict the output
y_pred #get the result
```

```
[[1 0]
 [2 0]
 [3 0]
 [6 0]]
[0 0 1 1]
array([1, 1, 1, 1])
```

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



```
from sklearn.metrics import confusion_matrix #importing confusion matrix package from skikit learn
confusion_matrix(y_test,y_pred) #implementing confusion matrix between tested and predicted data
```

```
array([[0, 3],
       [0, 1]])
```

Divya Sai Ajay Tasti
700741296

10)

Accuracy.

$$\text{Accuracy} = \frac{TN + TP}{T + W}$$

$$= \frac{0+1}{1+3} = \frac{1}{4} = 0.25$$

$$\text{Sensitivity} = \frac{TP}{FP+TP}.$$

$$= \frac{1}{0+1} = 1$$

$$\text{Specificity} = \frac{TN}{TN + FP}$$

$$= \frac{0}{0+3} = 0$$