

# ASSIGNMENT 1

```
#Question1
#list of ages
ages = [19, 22, 19, 24, 20, 25, 26, 24, 25, 24]
```

```
#Sorting the list ages(default)
ages.sort()
print("Sorted Ages=",ages)
```

```
#Min age from the list ages
min_age=min(ages)
print("Minimum_age:",min_age)
```

```
#Max age from the list ages
max_age=max(ages)
print("Maximum_age:",max_age)
```

```
#Add min age and max age to the list ages
ages.append(min_age)
ages.append(max_age)
print("Updated ages List:",ages)
```

```
mid=len(ages)//2
#Median age of the list ages
Med_age = (ages[mid] + ages[~mid]) / 2
print("Median of ages:",Med_age)
```

```
#Average age of the list ages
avg=sum(ages) / len(ages)
print("Average age:",avg)
```

```
#Range of the list ages
range=max(ages)-min(ages)
print("Range:",range)
```

The screenshot shows a Jupyter Notebook titled 'ML\_Assignment1\_700741049'. The code is as follows:

```
mid=len(ages)//2
#Median age of the List ages
Med_age = (ages[mid] + ages[-mid]) / 2
print("Median of ages:",Med_age)

#Average age of the list ages
avg=sum(ages) / len(ages)
print("Average age:",avg)

#Range of the list ages
range=max(ages)-min(ages)
print("Range:",range)

Sorted_Ages= [19, 19, 20, 22, 24, 24, 24, 25, 25, 26]
Minimum_age: 19
Maximum_age: 26
Updated ages List: [19, 19, 20, 22, 24, 24, 24, 25, 25, 26, 19, 26]
Median of ages: 24.0
Average age: 22.75
Range: 7

In [59]: #Question2
#Created empty dog dictionary
dog={}

#dog dictionary with keys and values
dog={'name':'charlie', 'color':'golden', 'breed':'goldenretriever', 'legs':'4', 'age':'5'}
print("Dog dictionary:",dog)

#Created Student dictionary
student = {
    'first_name':'Madhuri',
    'last_name':'Gadaboina',
    'gender':'Female',
    'age':'22',
    'country':'USA',
    'city':'overland park',
    'marital status':True,
    'skills':['JavaScript', 'React', 'Node', 'MongoDB', 'Python'],
    'address':{
        'street':'Space street',
        'zipcode':'02210'
    }
}
```

#Question2

#Created empty dog dictionary

dog={}

#dog dictionary with keys and values

dog={'name':'charlie', 'color':'golden', 'breed':'goldenretriever', 'legs':'4', 'age':'5'}  
print("Dog dictionary:",dog)

#Created Student dictionary

```
student = {
    'first_name':'Madhuri',
    'last_name':'Gadaboina',
    'gender':'Female',
    'age':'22',
    'country':'USA',
    'city':'overland park',
    'marital status':True,
    'skills':['JavaScript', 'React', 'Node', 'MongoDB', 'Python'],
    'address':{
        'street':'Space street',
        'zipcode':'02210'
    }
}
```

print("Student Dictionary:",student)

#length of Student Dictionary

```
print("Length of Student Dictionary:",len(student))
```

```
#Getting the value of skills
```

```
print("value of skills:",student['skills'])
```

```
#checking its data type
```

```
print("Datatype of Skills:",type(student['skills']))
```

```
#Added Values to Skills
```

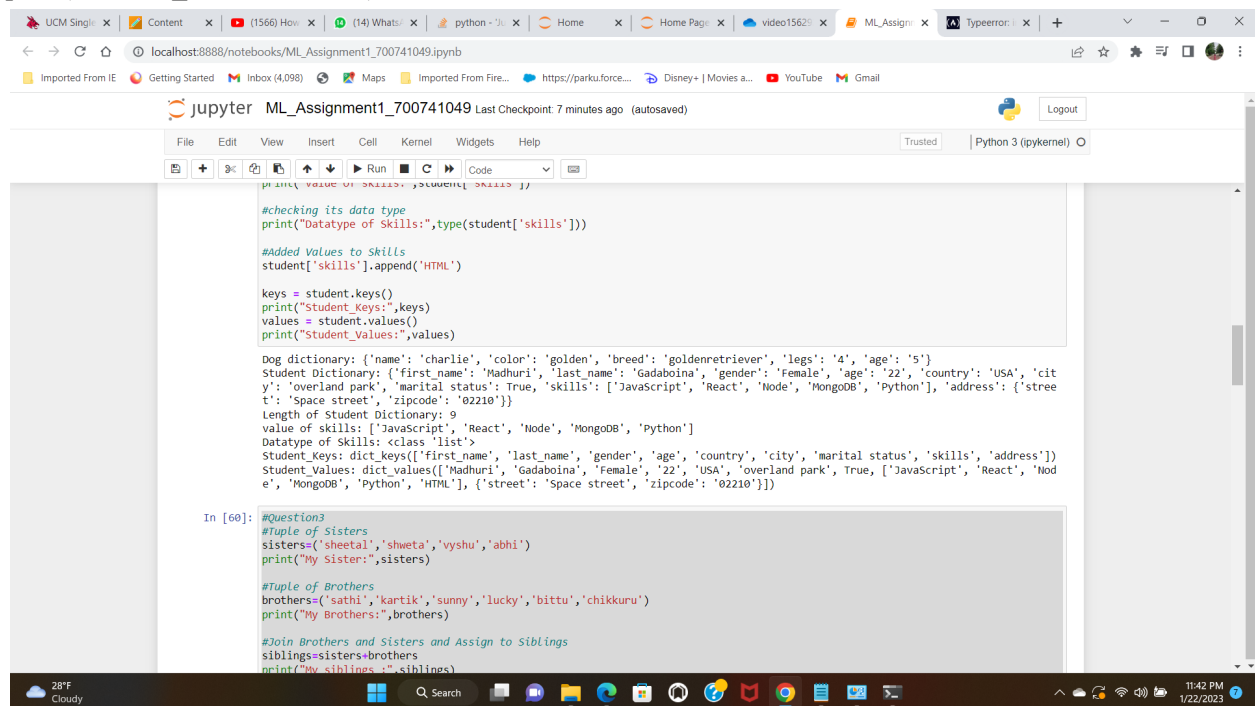
```
student['skills'].append('HTML')
```

```
keys = student.keys()
```

```
print("Student_Keys:",keys)
```

```
values = student.values()
```

```
print("Student_Values:",values)
```



```
print("value of skills:",student['skills'])

#checking its data type
print("Datatype of Skills:",type(student['skills']))

#Added Values to Skills
student['skills'].append('HTML')

keys = student.keys()
print("Student_Keys:",keys)
values = student.values()
print("Student_Values:",values)

Dog dictionary: {'name': 'charlie', 'color': 'golden', 'breed': 'goldenretriever', 'legs': '4', 'age': '5'}
Student Dictionary: {'first_name': 'Madhuri', 'last_name': 'Gadaboina', 'gender': 'Female', 'age': '22', 'country': 'USA', 'city': 'overland park', 'marital status': True, 'skills': ['JavaScript', 'React', 'Node', 'MongoDB', 'Python'], 'address': {'street': 'Space street', 'zipcode': '02210'}}
Length of Student Dictionary: 9
value of skills: ['JavaScript', 'React', 'Node', 'MongoDB', 'Python']
Datatype of Skills: <class 'list'>
Student_Keys: dict_keys(['first_name', 'last_name', 'gender', 'age', 'country', 'city', 'marital status', 'skills', 'address'])
Student_Values: dict_values(['Madhuri', 'Gadaboina', 'Female', '22', 'USA', 'overland park', True, ['JavaScript', 'React', 'Node', 'MongoDB', 'Python', 'HTML'], {'street': 'Space street', 'zipcode': '02210'}])

In [60]: #Question3
#Tuple of Sisters
sisters=('sheetal','shweta','vyshu','abhi')
print("My Sister:",sisters)

#Tuple of Brothers
brothers=('sathi','kartik','sunny','lucky','bittu','chikkuru')
print("My Brothers:",brothers)

#Join Brothers and Sisters and Assign to Siblings
siblings=sisters+brothers
print("My siblings :",siblings)
```

```
#Question3
```

```
#Tuple of Sisters
```

```
sisters=('sheetal','shweta','vyshu','abhi')
```

```
print("My Sister:",sisters)
```

```
#Tuple of Brothers
```

```
brothers=('sathi','kartik','sunny','lucky','bittu','chikkuru')
```

```
print("My Brothers:",brothers)
```

```
#Join Brothers and Sisters and Assign to Siblings
```

```
siblings=sisters+brothers
```

```
print("My siblings :",siblings)
```

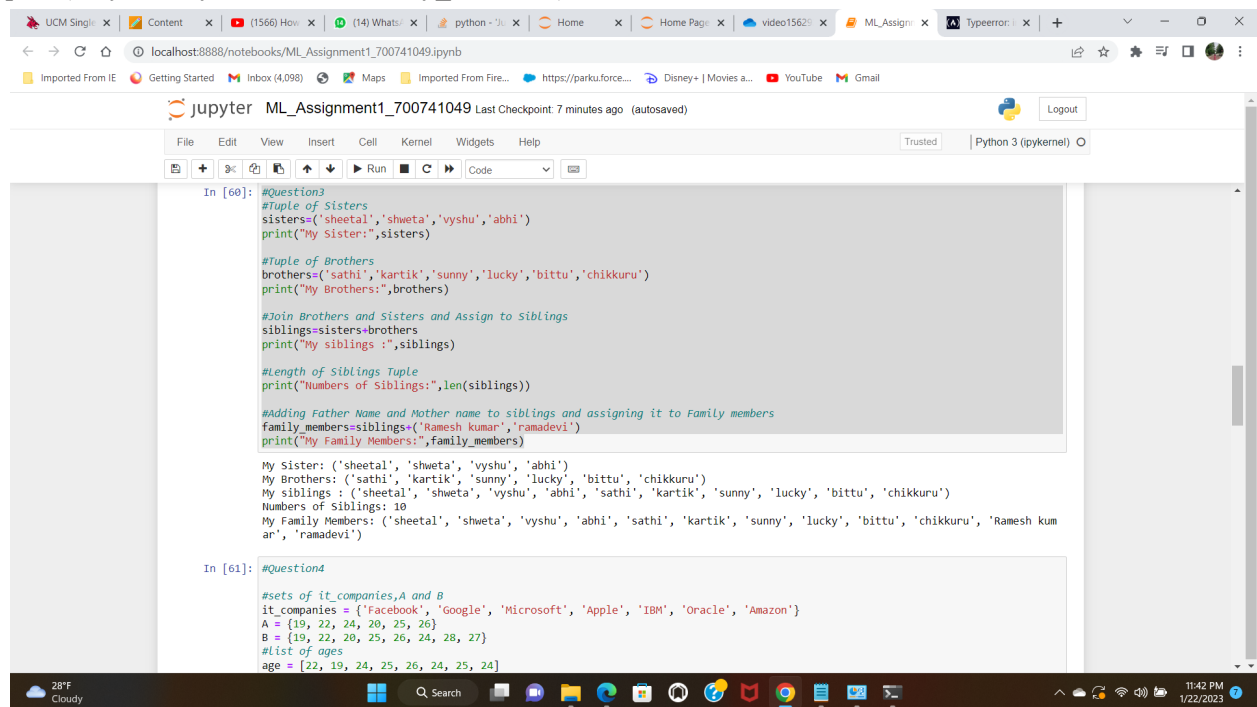
```
#Length of Siblings Tuple
```

```
print("Numbers of Siblings:",len(siblings))
```

```
#Adding Father Name and Mother name to siblings and assigning it to Family members
```

```
family_members=siblings+('Ramesh kumar','ramadevi')
```

```
print("My Family Members:",family_members)
```



```
#Question3
#Tuple of Sisters
sisters=('sheetal','shweta','vyshu','abhi')
print("My Sister:",sisters)

#Tuple of Brothers
brothers=('sathi','kartik','sunny','lucky','bittu','chikkuru')
print("My Brothers:",brothers)

#Join Brothers and Sisters and Assign to Siblings
siblings=sisters+brothers
print("My siblings :",siblings)

#Length of Siblings Tuple
print("Numbers of Siblings:",len(siblings))

#Adding Father Name and Mother name to siblings and assigning it to Family members
family_members=siblings+('Ramesh kumar','ramadevi')
print("My Family Members:",family_members)

My Sister: ('sheetal', 'shweta', 'vyshu', 'abhi')
My Brothers: ('sathi', 'kartik', 'sunny', 'lucky', 'bittu', 'chikkuru')
My siblings : ('sheetal', 'shweta', 'vyshu', 'abhi', 'sathi', 'kartik', 'sunny', 'lucky', 'bittu', 'chikkuru')
Numbers of Siblings: 10
My Family Members: ('sheetal', 'shweta', 'vyshu', 'abhi', 'sathi', 'kartik', 'sunny', 'lucky', 'bittu', 'chikkuru', 'Ramesh kumar', 'ramadevi')
```

```
#Question4
#sets of it_companies,A and B
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}
#List of ages
age = [22, 19, 24, 25, 26, 24, 25, 24]
```

```
#Question3
```

```
#Tuple of Sisters
```

```
sisters=('sheetal','shweta','vyshu','abhi')
```

```
print("My Sister:",sisters)
```

```
#Tuple of Brothers
```

```
brothers=('sathi','kartik','sunny','lucky','bittu','chikkuru')
```

```
print("My Brothers:",brothers)
```

```
#Join Brothers and Sisters and Assign to Siblings
```

```
siblings=sisters+brothers
```

```
print("My siblings :",siblings)
```

```
#Length of Siblings Tuple
```

```
print("Numbers of Siblings:",len(siblings))
```

```
#Adding Father Name and Mother name to siblings and assigning it to Family members
```

```
family_members=siblings+('Ramesh kumar','ramadevi')
```

```
print("My Family Members:",family_members)
```

The screenshot shows a Jupyter Notebook titled 'ML\_Assignment1\_700741049'. The code in the notebook is as follows:

```
Deleting all the sets
del A,B,it_companies

#converting list to set and finding their lengths
ages=set(age)
print(ages)
print("Length of set ages:",len(ages))
print("Length of list age:",len(age))

length of it_companies: 7
it_companies after adding Twitter: {'Google', 'Apple', 'Microsoft', 'Twitter', 'Oracle', 'Amazon', 'Facebook', 'IBM'}
updated it_companies: {'Apple', 'Twitter', 'Oracle', 'Facebook', 'Google', 'Cognizant', 'Amazon', 'IBM', 'Wipro', 'Microsoft', 'TCS'}
{'Apple', 'Twitter', 'Oracle', 'Facebook', 'Cognizant', 'Amazon', 'IBM', 'Wipro', 'Microsoft', 'TCS'}
Join of A and B: {19, 20, 22, 24, 25, 26, 27, 28}
Intersection of A and B: {19, 20, 22, 24, 25, 26}
A subset of B: True
A and B are Disjoint Sets: False
Symmetric Difference: {27, 28}
{19, 22, 24, 25, 26}
Length of set ages: 5
Length of list age: 8

In [62]: #Question5
import math as M

#radius as user input
Radius = float(input("Please enter the radius of the given circle: "))

#area of circle
_area_of_the_circle_ = M.pi* Radius * Radius
print(" The area of the given circle = %.2f" % _area_of_the_circle_)

#Circumference of the circle
_circum_of_circle_ = 2 * M.pi * radius
print(" Circumference Of a Circle = %.2f" % _circum_of_circle_)
```

#Question5

```
import math as M
```

#radius as user input

```
Radius = float(input("Please enter the radius of the given circle: "))
```

#area of circle

```
_area_of_the_circle_ = M.pi* Radius * Radius
```

```
print(" The area of the given circle = %.2f" % _area_of_the_circle_)
```

#Circumference of the circle

```
_circum_of_circle_ = 2 * M.pi * radius
```

```
print(" Circumference Of a Circle = %.2f" % _circum_of_circle_)
```

The screenshot shows a Jupyter Notebook with the following code and output:

```
import math as M

#radius as user input
Radius = float(input("Please enter the radius of the given circle: "))

#area of circle
_area_of_the_circle = M.pi* Radius * Radius
print("The area of the given circle = %.2f" % _area_of_the_circle_)

#Circumference of the circle
_circum_of_circle = 2 * M.pi * radius
print("Circumference Of a Circle = %.2f" % _circum_of_circle_)

Please enter the radius of the given circle: 10
The area of the given circle = 314.16
Circumference Of a Circle = 62.83

In [63]: #Question6
string = "I am a teacher and I love to inspire and teach people"
# Use split method to separate the words and set to get the unique values
uniq=set(string.split(" "))
print("Unique Words:",uniq)
print("Length:",len(uniq))

Unique Words: {'and', 'a', 'I', 'inspire', 'teach', 'to', 'love', 'teacher', 'people', 'am'}
Length: 10

In [64]: #Question7
#used escape Sequence tab
print("Name\tAge\tCountry\tCity\t\nAsabeneh 250\tFinland\tHelsinki")

Name    Age    Country City
```

#Question6

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

# Use split method to separate the words and set to get the unique values

uniq=set(string.split(" "))

print("Unique Words:",uniq)

print("Length:",len(uniq))

The screenshot shows a Jupyter Notebook with the following code and output:

```
#Deleting all the sets
del A,B,it_companies

#converting list to set and finding their lengths
ages=set(age)
print(ages)
print("Length of set ages:",len(ages))
print("Length of list age:",len(age))

length of it_companies: 7
it_companies after adding Twitter: {'Google', 'Apple', 'Microsoft', 'Twitter', 'Oracle', 'Amazon', 'Facebook', 'IBM'}
updated it_companies: {'Apple', 'Twitter', 'Oracle', 'Facebook', 'Google', 'Cognizant', 'Amazon', 'IBM', 'Wipro', 'Microsoft', 'TCS'}
{'Apple', 'Twitter', 'Oracle', 'Facebook', 'Cognizant', 'Amazon', 'IBM', 'Wipro', 'Microsoft', 'TCS'}
Join of A and B: {19, 20, 22, 24, 25, 26, 27, 28}
Intersection of A and B: {19, 20, 22, 24, 25, 26}
A subset of B: True
A and B are Disjoint Sets: False
Symmetric Difference: {27, 28}
{19, 22, 24, 25, 26}
Length of set ages: 5
Length of list age: 8

In [62]: #Questions
import math as M

#radius as user input
Radius = float(input("Please enter the radius of the given circle: "))

#area of circle
_area_of_the_circle = M.pi* Radius * Radius
print("The area of the given circle = %.2f" % _area_of_the_circle_)

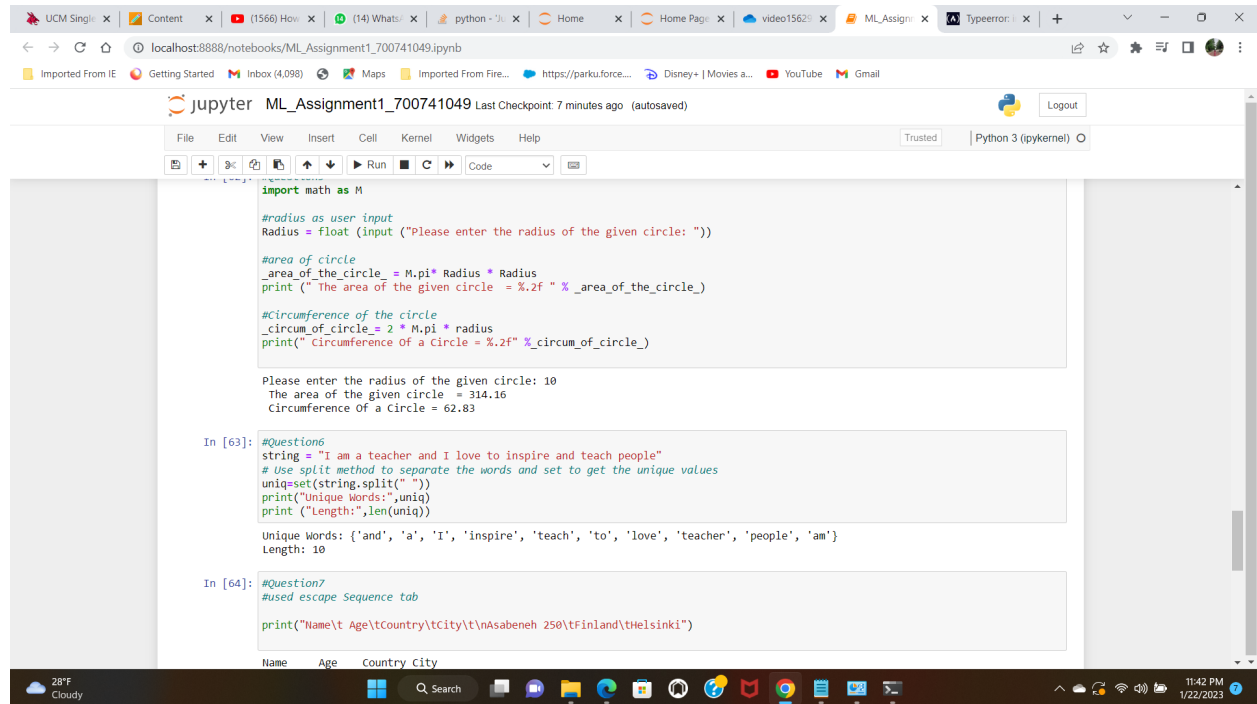
#Circumference of the circle
_circum_of_circle = 2 * M.pi * radius
print("Circumference of a Circle = %.2f" % _circum_of_circle_)

Please enter the radius of the given circle: 10
The area of the given circle = 314.16
Circumference of a Circle = 62.83
```

#Question7

#used escape Sequence tab

```
print("Name\t Age\tCountry\tCity\t\nAsabeneh 250\tFinland\tHelsinki")
```



The screenshot shows a Jupyter Notebook window titled 'ML\_Assignment1\_700741049'. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, running, and code execution. The notebook contains three code cells. The first cell calculates the area and circumference of a circle based on user input. The second cell processes a string to find unique words and their length. The third cell uses escape sequences to print a formatted string. The output of the first cell shows the user input '10' and the calculated area '314.16' and circumference '62.83'. The output of the second cell shows the unique words and their length '10'. The output of the third cell shows the formatted string with tabs.

```
import math as M

#radius as user input
Radius = float(input("Please enter the radius of the given circle: "))

#area of circle
_area_of_the_circle = M.pi* Radius * Radius
print("The area of the given circle = %.2f" % _area_of_the_circle_)

#Circumference of the circle
_circum_of_circle = 2 * M.pi * radius
print("Circumference Of a Circle = %.2f" % _circum_of_circle_)

Please enter the radius of the given circle: 10
The area of the given circle = 314.16
Circumference Of a Circle = 62.83

In [63]: #Question6
string = "I am a teacher and I love to inspire and teach people"
# Use split method to separate the words and set to get the unique values
uniq=set(string.split(" "))
print("Unique Words:",uniq)
print ("Length:",len(uniq))

Unique Words: {'and', 'a', 'I', 'inspire', 'teach', 'to', 'love', 'teacher', 'people', 'am'}
Length: 10

In [64]: #Question7
#used escape Sequence tab

print("Name\t Age\tCountry\tCity\t\nAsabeneh 250\tFinland\tHelsinki")

Name    Age    Country City
Asabeneh 250 Finland Helsinki
```

#Question8

#Using String format method

```
print(fradius = 10')
```

```
print(f'area = 3.14*radius**2')
```

```
print(f"The area of circle with radius {r} is {3.14*r*r} meters square")
```

```
#used escape Sequence tab
print("Name\t Age\tCountry\tCity\t\nAsabeneh 250\tFinland\tHelsinki")

Name    Age    Country City
Asabeneh 250    Finland Helsinki

In [65]: #Question8
#Using string format method
print(f'radius = 10')
print(f'area = 3.14*radius**2')
print(f"The area of circle with radius {r} is {3.14*r*r} meters square")

radius = 10
area = 3.14*radius**2
"The area of circle with radius 7 is 153.86 meters square"

In [66]: #Question9
#Creating a list(L1) for weights(lbs) of N students
L1=[int(num) for num in input().split(" ")]
#Creating another list called W_kg
W_kg=[]
#Using for loop to iterate the values and appending the list
for i in L1:
    W_kg.append(round(i/2.205,2))
#Displaying the values in kgs after conversion
print ("Values are:",W_kg)

155 150 145 148
Values are: [70.29, 68.03, 65.76, 67.12]
```

#Question9

#Creating a list(L1) for weights(lbs) of N students

L1=[int(num) for num in input().split(" ")]

#Creating another list called W\_kg

W\_kg=[]

#Using for loop to iterate the values and appending the list

for i in L1:

W\_kg.append(round(i/2.205,2))

#Displaying the values in kgs after conversion

print ("Values are:",W\_kg)

10



10

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

1) Using KNN classifier where  $K=3$

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

$(6, 6)$   $(6, 3)$   $(6, 2)$   $(6, 1)$  are points need to be calculated

$$x = (0, 0, 1)$$

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

$$d = \sqrt{(6-3)^2} = \sqrt{9} = 3$$

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

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

Max = 0 calculate for rest points i.e., 0

2) Confusion Matrix

$$\text{Accuracy} = (TP + TN) / (TN + FP + FN + TP)$$

$$= (0 + 1) / (1 + 0 + 3 + 0) = 1/4$$

$$\text{Sensitivity} = TP / (TP + FN) = 0 / (0 + 3) = 0$$

$$\text{Specificity} = TN / (FP + TN) = 1 / 0 + 1 = 1$$

	0	1
0	TN = 1	FP = 0
1	FN = 3	TP = 0

The screenshot displays a Jupyter Notebook environment. The top bar shows the URL `localhost:8888/notebooks/ML_Assignment1_700741049.ipynb`. The notebook title is `ML_Assignment1_700741049`, and it indicates the last checkpoint was 7 minutes ago (autosaved). The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for saving, undo, redo, and running code. The code editor shows two cells. The first cell contains a Python script for calculating the area of a circle with a radius of 10. The second cell contains a Python script for creating a list of student weights, converting them to kilograms, and displaying the results. The output of the second cell is visible, showing the list of values and their conversion to kilograms.

```
print(r"area = 3.14*radius**2")
print(f"The area of circle with radius {r} is {3.14*r*r} meters square")

radius = 10
area = 3.14*radius**2
"The area of circle with radius 7 is 153.86 meters square"
```

```
In [66]: #Question9
#Creating a list(L1) for weights(lbs) of N students
L1=[int(num) for num in input().split(" ")]
#Creating another list called W_kg
W_kg=[]
#Using for loop to iterate the values and appending the list
for i in L1:
    W_kg.append(round(i/2.205,2))
#Displaying the values in kgs after conversion
print ("Values are:",W_kg)

155 150 145 148
Values are: [70.29, 68.03, 65.76, 67.12]
```

The bottom of the notebook shows the input prompt `In [ ]:` for the next cell. The system tray at the bottom indicates the temperature is 28°F and the date is 1/22/2023.