**Shreshta Gottiparthi**

**700746153**

**Assignment-01**

**Google Drive link(Video link)** - <https://drive.google.com/file/d/1rfqHgJ-_dLxynvSKxDQlhGJSU7Hhks_6/view?usp=share_link>

**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

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

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

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

• Find the range of the ages (max minus min)

**Solution:**

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

#Sorting the above list - using sort() method, we can directly sort the list in ascending or descending order.

ages.sort()

print("List after sorting", ages)

#Finding min and max ages - using min() and max() functions, we can find the min and max ages

x = min(ages) #storing minimum age in x

y = max(ages) #storing maximum age in y

print("Min age is:", x)

print("Max age is:", y)

#Adding min and max age again to the list - using append() method we can add the elements to the list.

ages.append(x)

ages.append(y)

print("After adding min and max ages to the list:", ages)

#After adding the ages again, I have again sorted the list for findind the median

ages.sort()

print("List after sorting again after adding ages", ages)

#Finding the median age

#getting length of ages list

a = len(ages)

#if-else condition to check length is even or odd

#if even, take avg of middle two numbers as median else

if(a%2 == 0):

median = (ages[a//2] + ages[a//2-1])/2 #if even, take avg of middle two numbers as median

else:

median = ages[n//2] #if odd, take middle number as median

print("Median is:", median)

#Finding the average age - sum of all ages divided by length of all ages.

average = sum(ages)/a

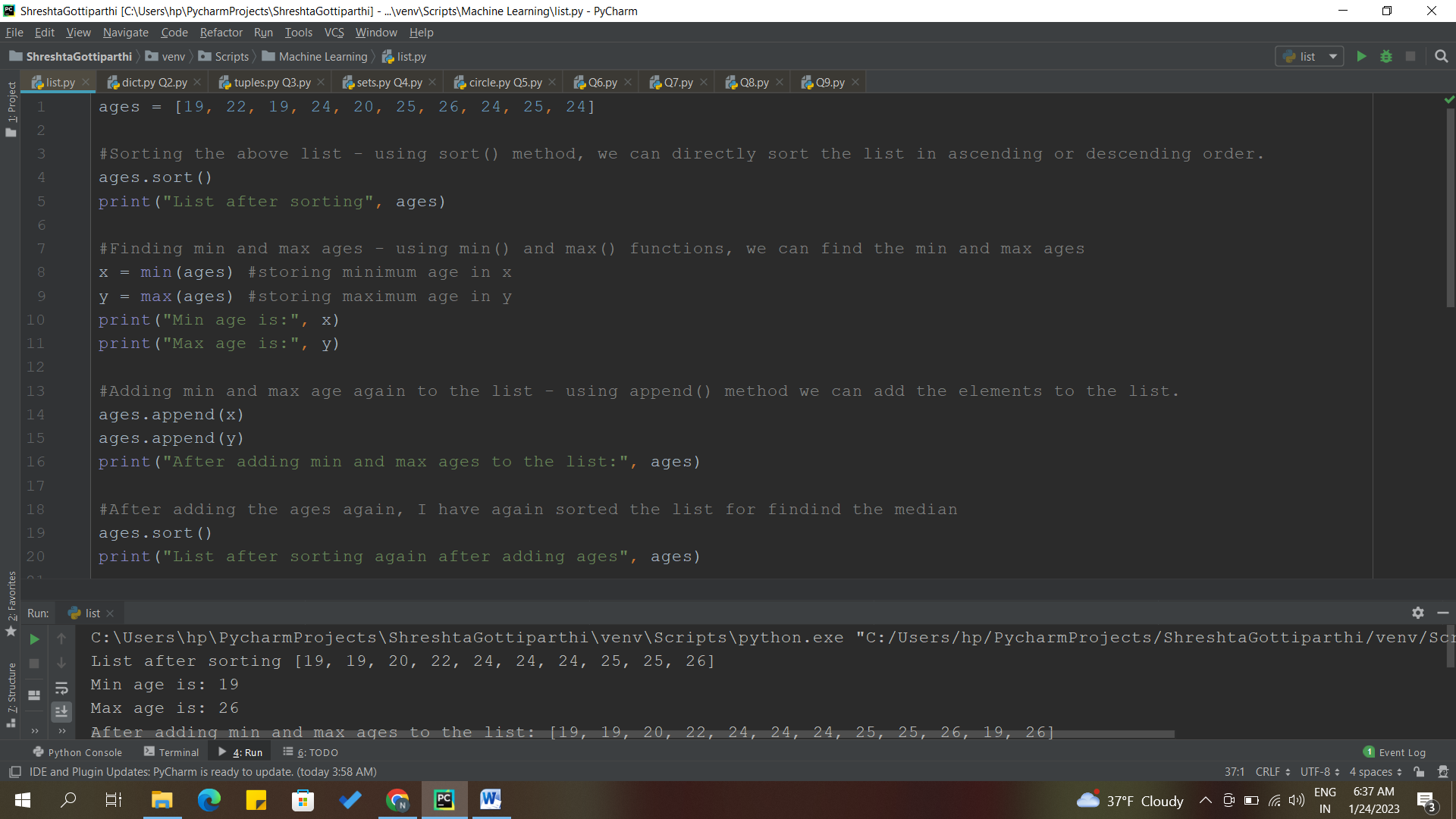
print("Average age:", average)

#Finding the range of ages - Subtracting the max age with min age.

print("Range of ages:", y-x)

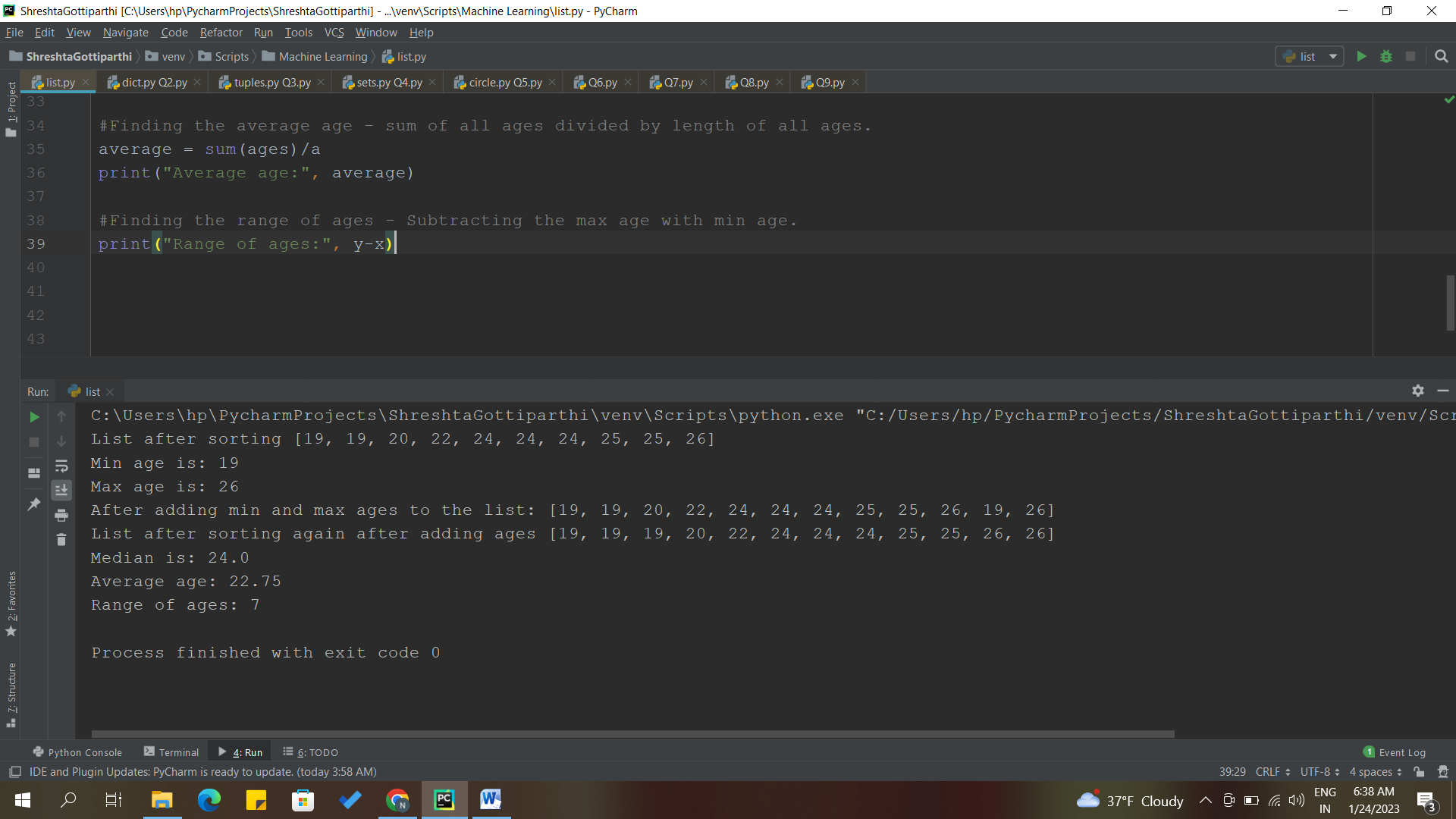
**Screenshots:**

**CODE:**





**OUTPUT:**

****

**DESCRIPTION:**

* Sorted the given list using sort(). By default, it sorts the list in ascending order.
* After that, I have stored min age in x and max age in y.
* Found the min and max ages using min() and max() functions.
* Adding min and max ages again to the list using append() method.
* Sorted the list again for finding the median of ages.
* Before finding the median, we need to get the length of ages list. For that, we use len() function.
* Used if-else condition to check if the length is even or odd.
* If even, take average of middle two numbers as median.
* Else, take middle number as median.
* For finding the average age, we take the sum of all ages divided by length of all ages.
* We subtract the min age from the max age to get the range of ages.

**Question 2**

• Create an empty dictionary called dog

• Add name, color, breed, legs, age to the dog dictionary

• Create a student dictionary and add first\_name, last\_name, gender, age, marital status, skills, country, city and address as keys for the dictionary

• Get the length of the student dictionary

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

• Modify the skills values by adding one or two skills

• Get the dictionary keys as a list

• Get the dictionary values as a list

**Solution:**

#Creating an empty dictionary called dog - Dictionaries are represented as {} with key:value pairs.

dog = {}

print("Empty Dictionary") #printing an empty dictionary

#Need to mention the key value pairs in " ", if they are strings.

dog["name"] = "Suri" #adding the key as name and value as Suri to the dictionary

dog["color"] = "White" #adding the key as color and value as White to the dictionary

dog["breed"] = "Shih Tzu" #adding the key as breed and value as Shih Tzu to the dictionary

dog["legs"] = 4 #adding the key as legs and value as 4 to the dictionary

dog["age"] = 2 #adding key as age and value as 2 to the dictionary

print(dog) #Printing the dictionary after adding the key:value pairs.

Student\_dict = {"first\_name": "Shreshta", "last\_name": "Gottiparthi", "gender": "F", "age": 23,

"marital\_status": "Single", "skills":["C", "Python"], "country": "India",

"city": "Hyderabad", "address": "Himayatnagar, Telangana"}

print(Student\_dict)

l = len(Student\_dict) #len() is used to get the length.

print("Length of Student\_dict:", l) #we get the length of student dictionary

print("Items in skills:", Student\_dict["skills"]) #this is to get the value of skills present in the student dictionary.

print("Type of Student\_dict:", type(Student\_dict["skills"])) #type() is used to check the datatype. Checking the type of skills in the student

#dictionary and printing the same.

Student\_dict["skills"].append("SQL") #adding an extra element in the skills present in Student\_dict

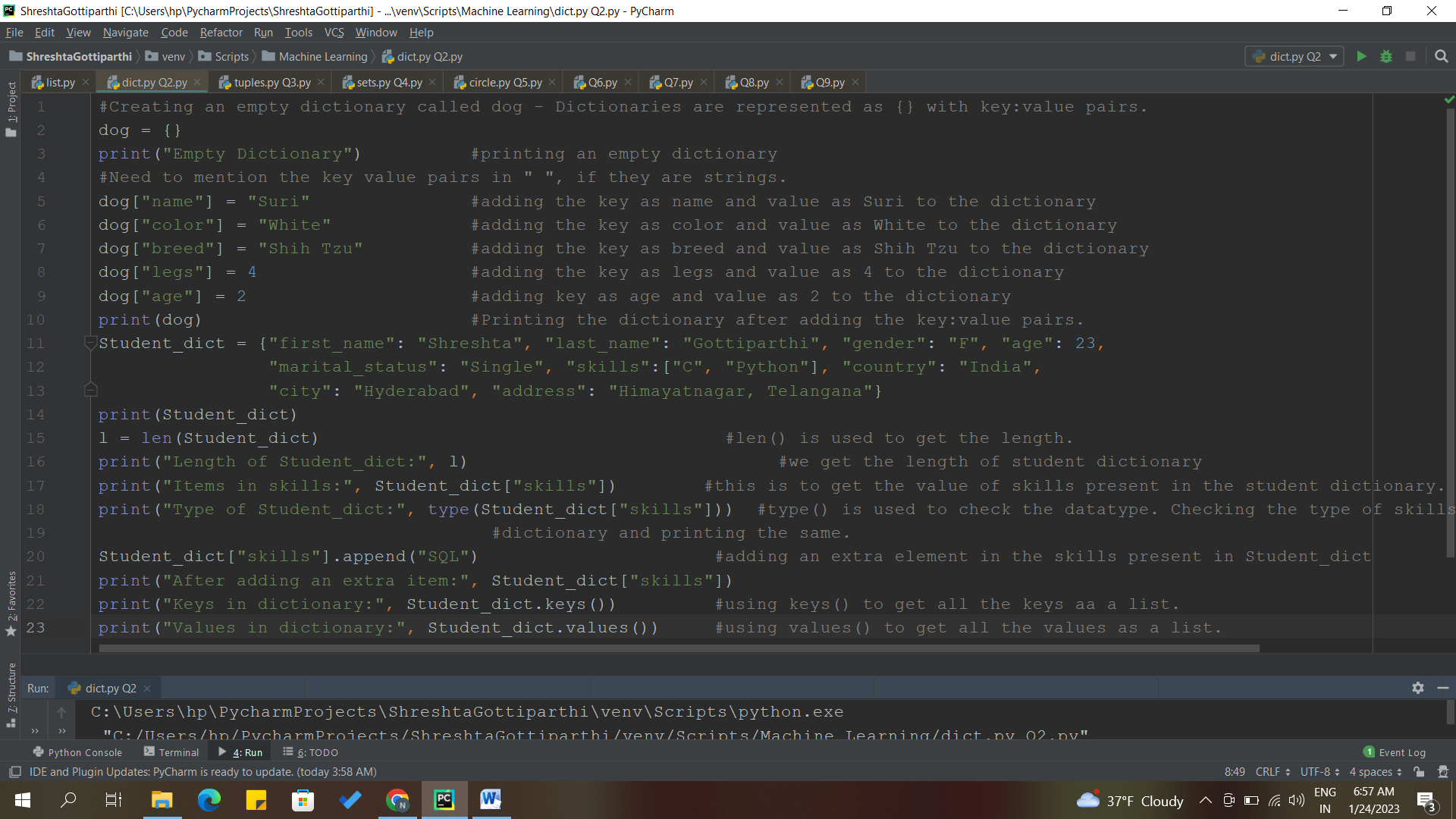
print("After adding an extra item:", Student\_dict["skills"])

print("Keys in dictionary:", Student\_dict.keys()) #using keys() to get all the keys aa a list.

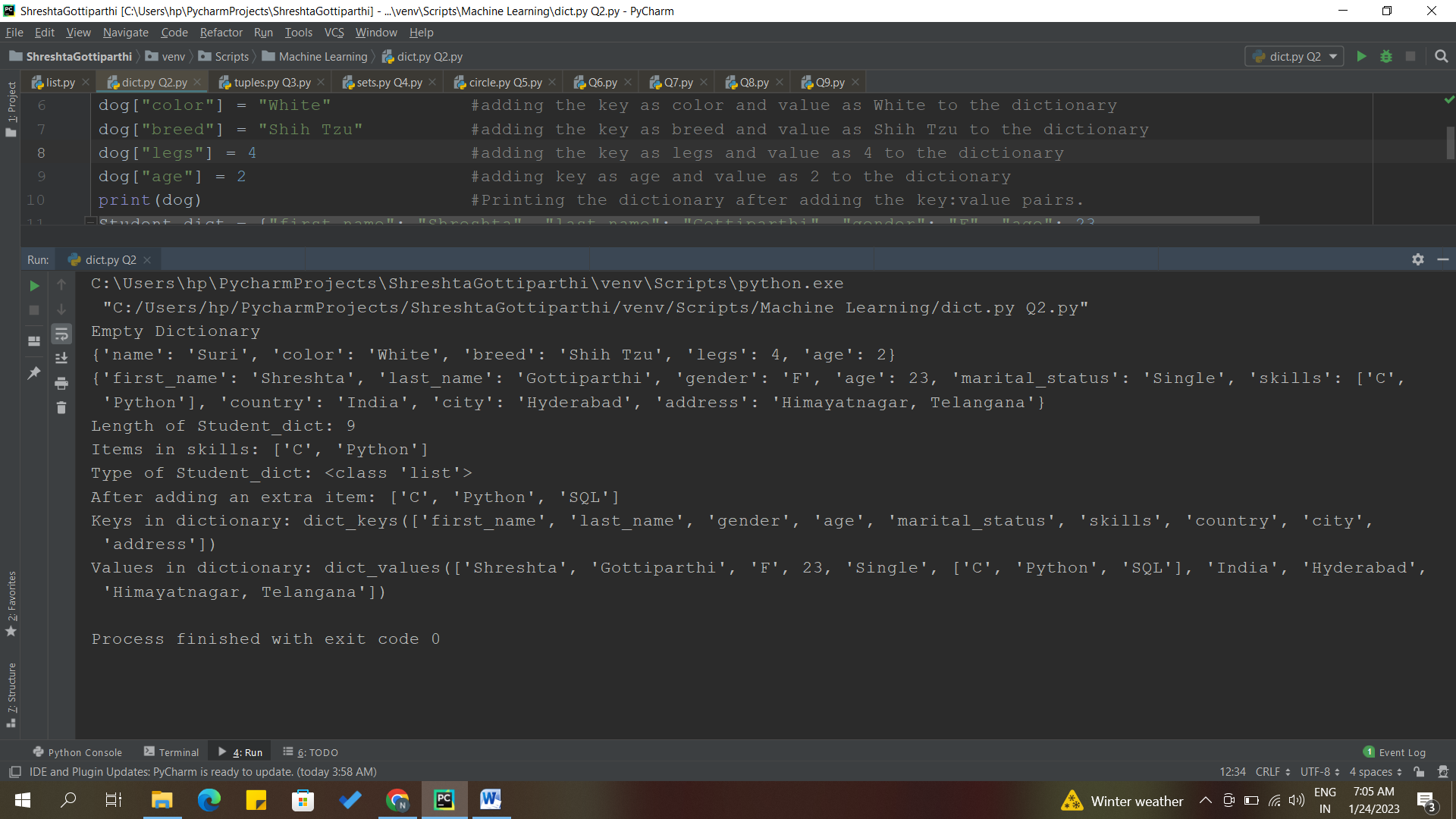
print("Values in dictionary:", Student\_dict.values()) #using values() to get all the values as a list.

**Screenshots:**

**CODE:**

****

**OUTPUT:**



**DESCRIPTION:**

* Created an empty dictionary called dog. Dictionaries are represented as {} with key: value pairs.
* Adding key value pairs to the dog dictionary.
* Created a student dictionary and added the given items as keys and have put the values to it.
* To get the length of student dictionary, we use len() function.
* To check the values in skills, we write it as Student\_dict["skills"].
* To check the type of skills, we use type() function.
* To add an element in skills list, we use append() method.
* To get the keys and values separately as a list, we use keys() and values() methods.

**Question 3**

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

• Join brothers and sisters tuples and assign it to siblings

• How many siblings do you have?

• Modify the siblings tuple and add the name of your father and mother and assign it to family\_members

**Solution:**

#Creating a tuple containing names of sisters and brothers.

sisters =("Sneha", "Neha")

brothers = ("Rahul", "Varun", "Rohan")

#Joining the above two tuples and assigning it to a new tuple called siblings.

siblings = sisters+brothers #we use + to combine both the tuples.

print("Siblings are:", siblings)

s = len(siblings) #using len() to check the number of siblings present in the siblings tuple.

print("Number of siblings:", s)

#Tuples are immutable. We cannot modify a tuple!!

family\_members = siblings+("Geeta","Venu") #So here, I'm taking whole siblings tuple and adding the name of mother and father, assigning it to a new tuple called family\_members.

print("Family Members:", family\_members)

**Screenshots:**

**CODE AND OUTPUT:**

****

**DESCRIPTION:**

* Created two tuples named sisters and brothers.
* Joined both the tuples using + operator and named it as sibllings.
* To find the number of siblings, use len() function.
* We have been asked to Modify the siblings tuple and add the name of your father and mother and assign it to family\_members but Tuples are immutable. We cannot modify a tuple.
* So, I have taken whole siblings tuple and added the names of mother and father, assigned it to a new tuple called family\_members.

**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

• 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

• What is the difference between remove and discard

• 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.

**Solution:**

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]

x = len(it\_companies) #to get the length of it\_companies set, we use #len().

print("Length of the it\_companies:", x)

#Adding 'Twitter' to it\_companies set.

it\_companies.add("Twitter") #we use add() to add any element to a set.

print(it\_companies)

#Inserting multiple IT companies at once to the set it\_companies

it\_companies.update(["Flipkart", "HP", "Wipro"]) #for adding multiple elements #to a set, we use update().

print(it\_companies)

#Removing one of the companies from the set it\_companies

it\_companies.remove("IBM") #use remove() for removing an #element from the set.

print(it\_companies)

#What is the difference between remove and discard

#The difference between remove() and dicard() is that the disacrd() method will #not display any error if the specified item is not in the set. Whereas, remove() #method will display an error.

#Joining A and B

C = A.union(B) #using union() method we get #all the elements from both the sets.

print("After joining A and B:", C)

#Finding A intersection B

D = A.intersection(B) #returns a set which are #common in both the sets A and B.

print("Intersection of A and B:", D)

#Is A subset of B

print("Is A subset of B?", A.issubset(B)) #checks if A is a subset of B, #returns true if it is a subset else, returns false.

#Are A and B disjoint sets

print("Are A and B disjoint sets?", A.isdisjoint(B)) #checks if A and B are #disjoints,if yes returns true else, false.

#Join A with B and B with A

X = A.copy() #copying the A set to X.

A.update(B) #inserting all the ietms in B set to A set.

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

B.update(X) #inserting all the items in X set to B set.

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

#What is the symmetric difference between A and B

A = {19, 22, 24, 20, 25, 26}

B = {19, 22, 20, 25, 26, 24, 28, 27}

Y = A.symmetric\_difference(B) #returns the set in which the items are #not present in both the sets.

print("Symmetric Difference:", Y)

#Deleting the sets completely

del A

del B

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

ages = set(age) #converting the given list age to a #set called ages using set().

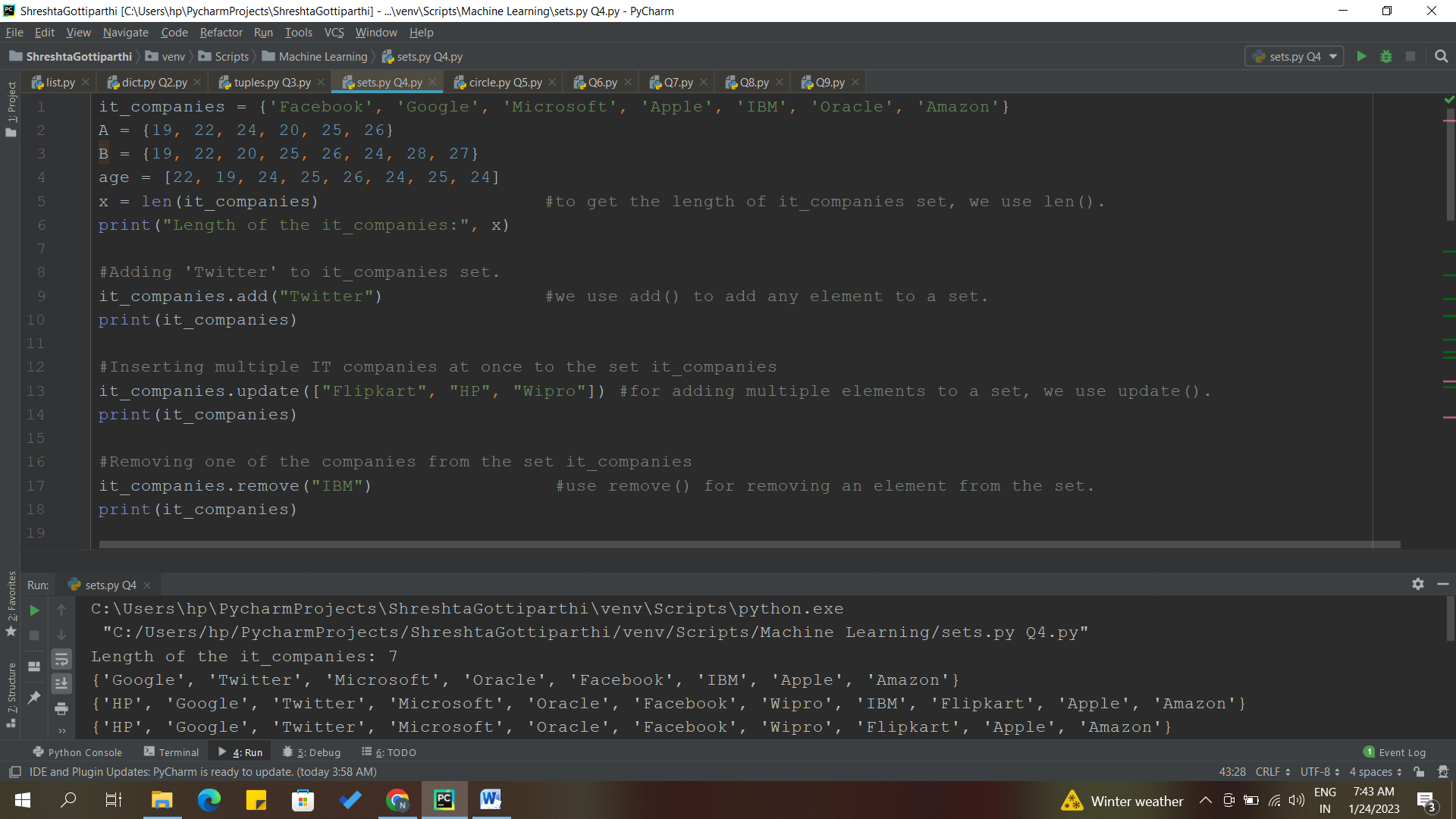
print("After converting list to a set:", ages)

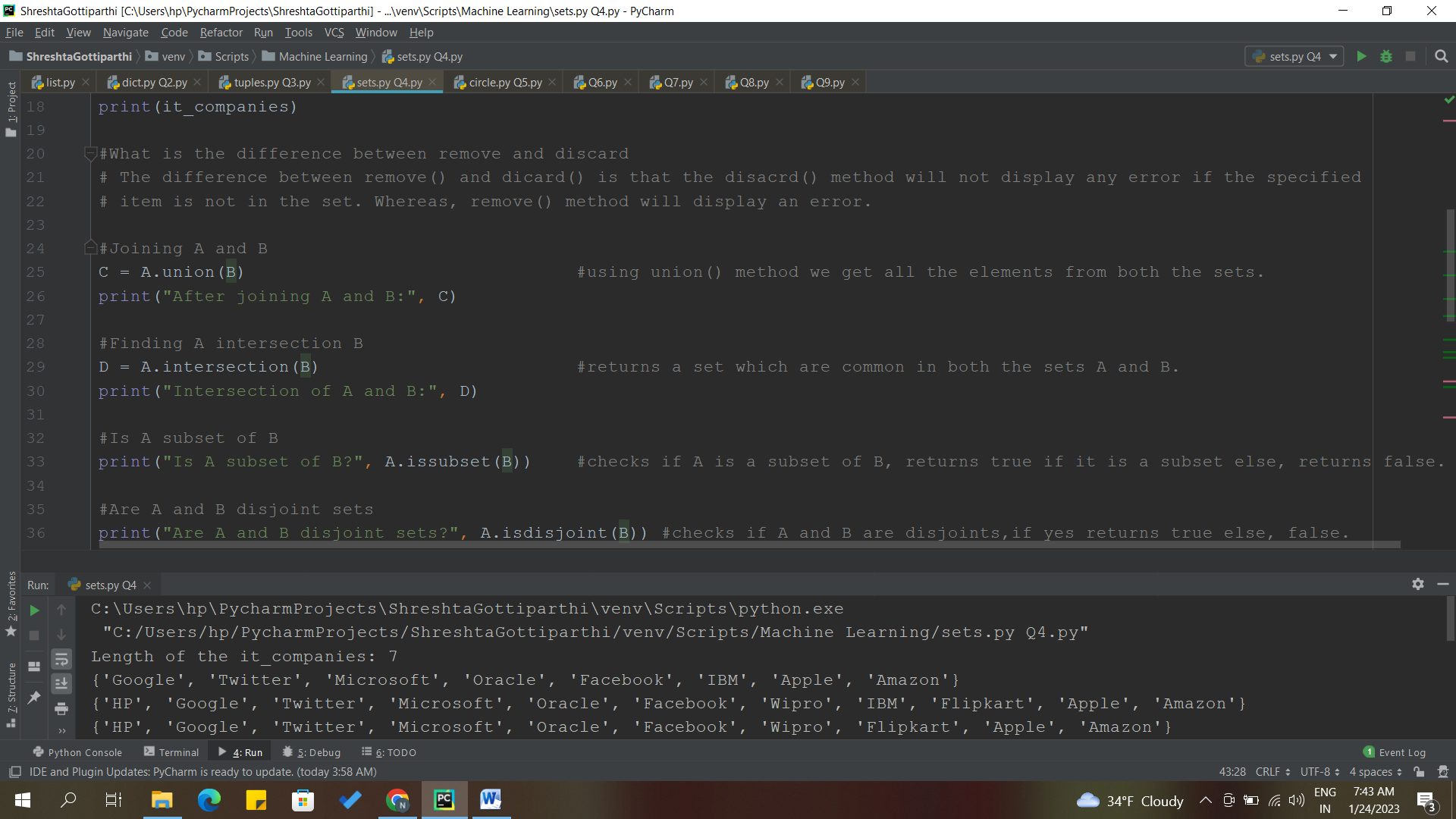
print("Length of a list- age:", len(age)) #checking the length of a list.

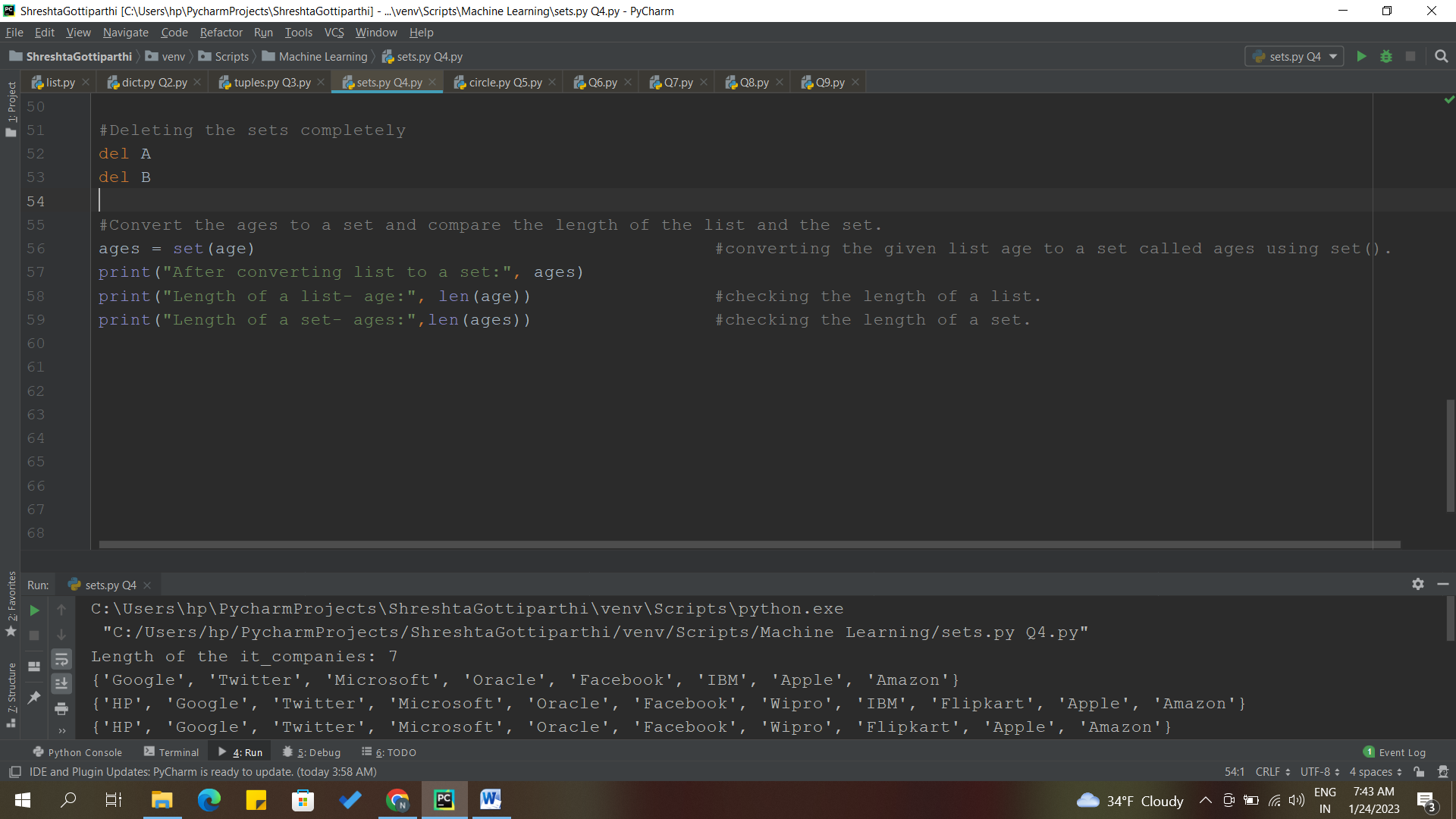
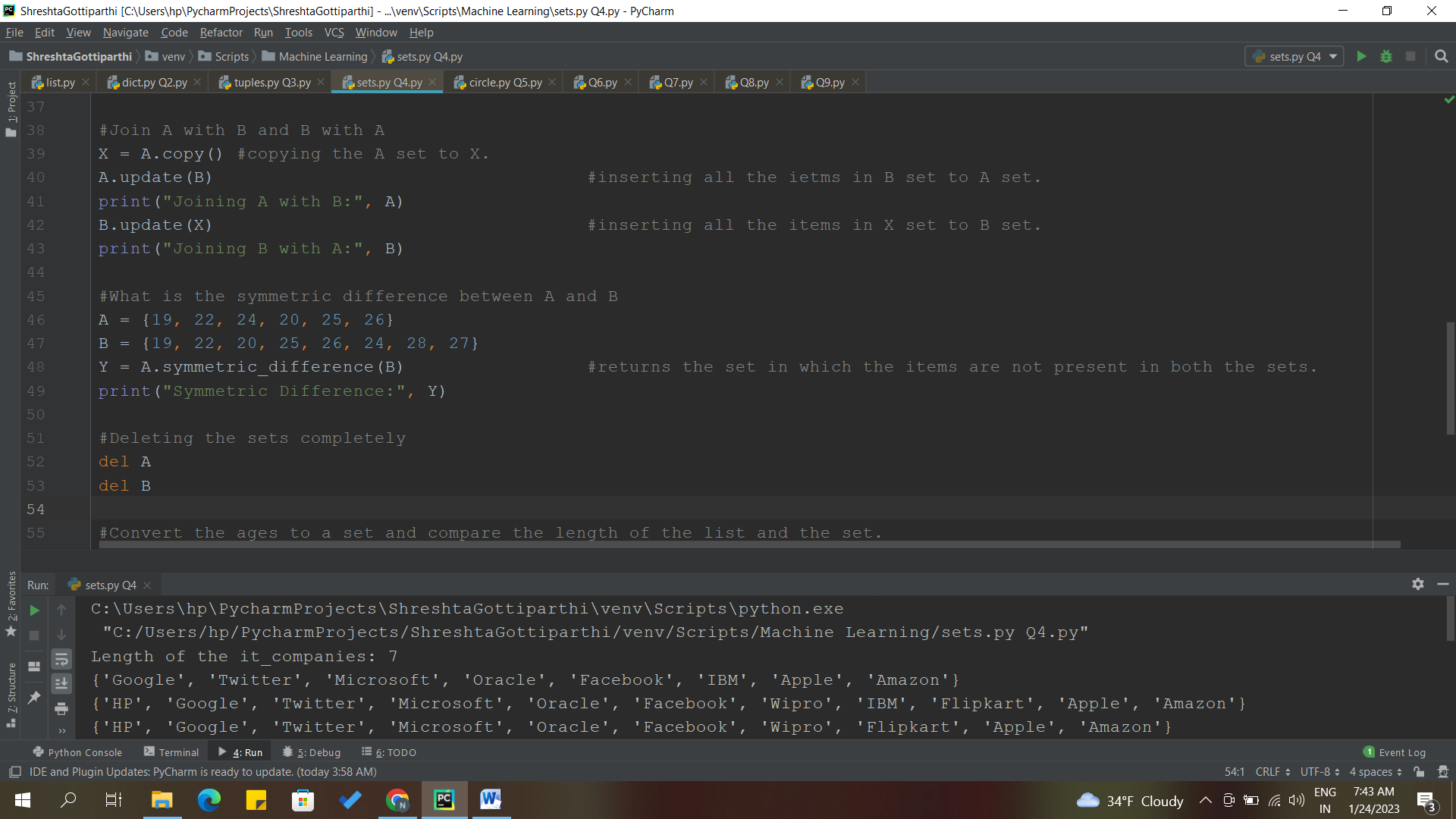
print("Length of a set- ages:",len(ages)) #checking the length of a set.

**Screenshots:**

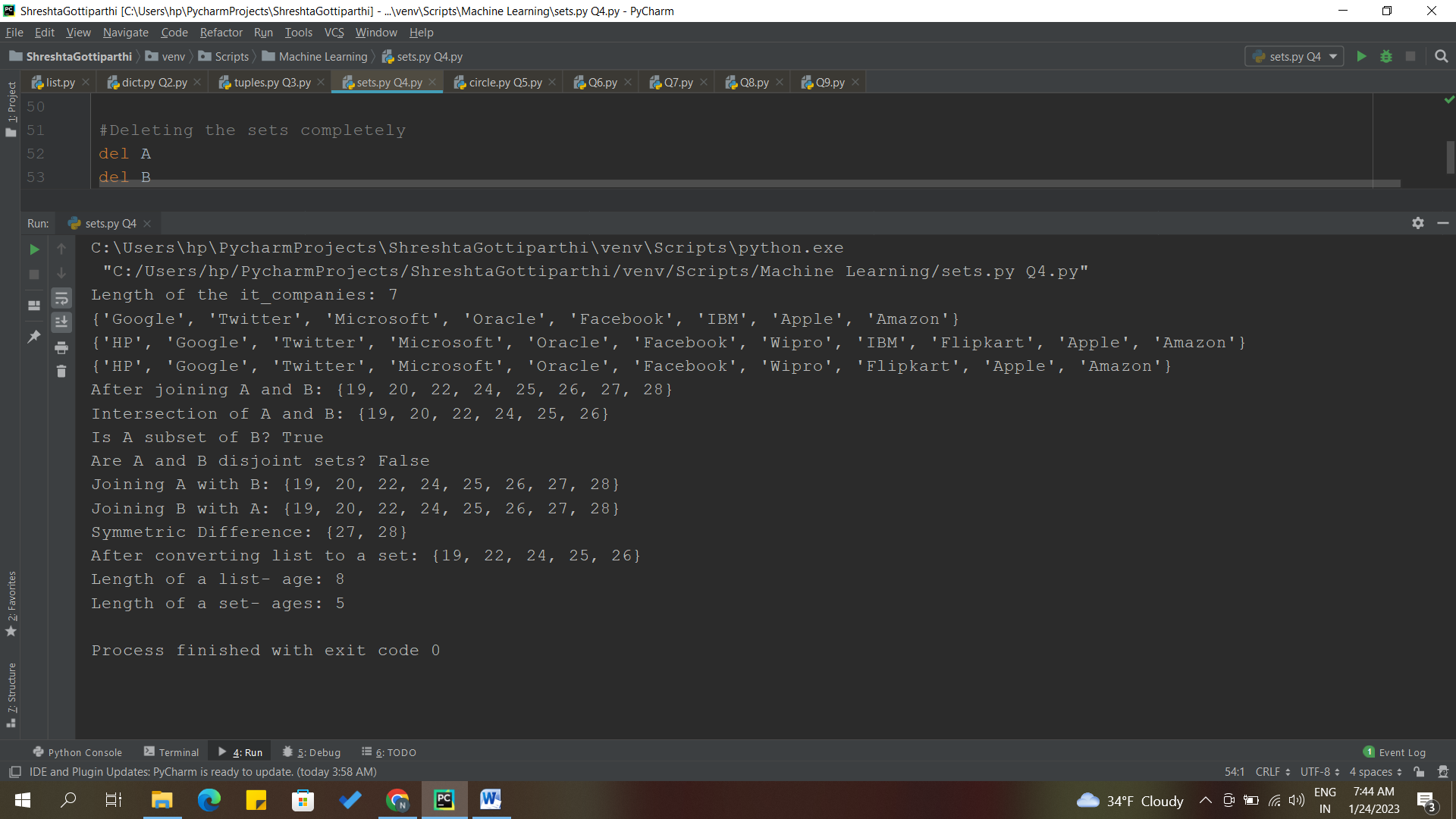
**CODE:**

****

****

****

**OUTPUT:**

****

**DESCRIPTION:**

* To check the length of a set, we use len() function.
* Added “Twitter” to the given set it\_companies using add() method.
* For adding multiple it companies, used update() method.
* #What is the difference between remove and discard

# The difference between remove() and dicard() is that the disacrd() method will not display any error if the specified item is not in the set. Whereas, remove() method will display an error.

* To remove an IT company from the set, used remove() method.
* For joining and intersecting two sets, we use union() which displays all the items from both the sets and intersection() which returns the common items from both the sets.
* For checking if one set is a subset of other set and if two sets are disjoint sets or not, we use issubset() and isdisjoint() methods.
* Issubset - checks if A is a subset of B, returns true if it is a subset else, returns false.
* Isdisjoint - checks if A and B are disjoints,if yes returns true else, false.
* Copied set A to a temp variable ‘X’.
* Joining A with B and B with A.
* For symmetric difference of two sets, used symmetric\_difference() method, returns the set in which the items are not present in both the sets.
* To delete a set, used ‘del’ and the set name.
* Converted age list to set using set().
* Finally, compared the length of the list and set, length of list is 8 and length of set is 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\_

• 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.

**Solution:**

radius = 30

\_area\_of\_circle\_ = 3.14\*radius\*\*2 #calculating area of a circle and assigning it to a variable \_area\_of\_circle\_

print("The area of a circle is:", \_area\_of\_circle\_)

\_circum\_of\_circle\_ = 2\*3.14\*radius #calculating circumference of a circle and assigning it to a variable \_circum\_of\_circle\_

print("The circumference of a circle is:", \_circum\_of\_circle\_)

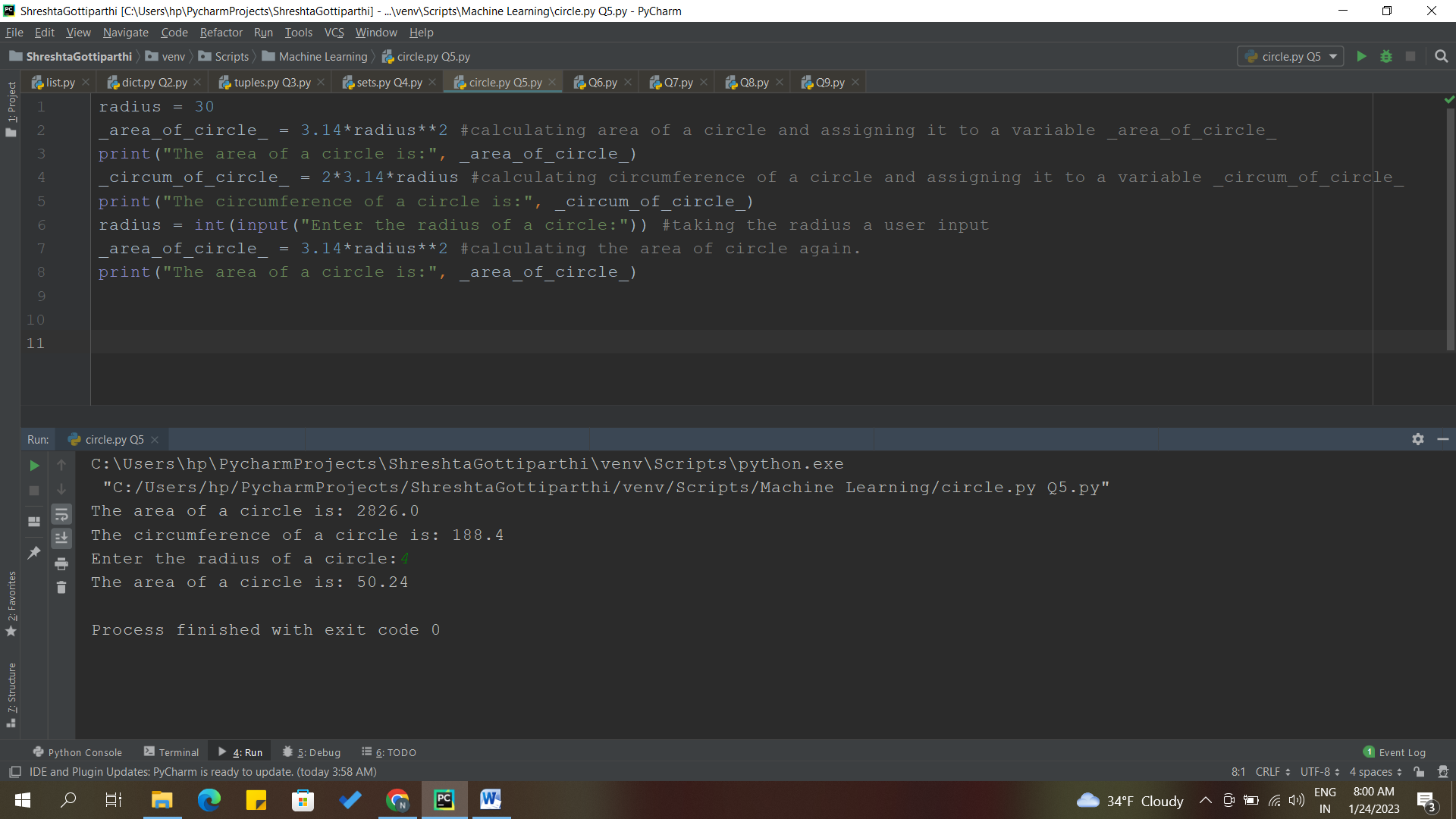
radius = int(input("Enter the radius of a circle:")) #taking the radius a user input

\_area\_of\_circle\_ = 3.14\*radius\*\*2 #calculating the area of circle again.

print("The area of a circle is:", \_area\_of\_circle\_)

**Screenshots:**

**CODE AND OUTPUT:**

****

**DESCRIPTION:**

Given radius of circle = 30

* Calculated area of a circle and assigned it to a variable \_area\_of\_circle\_ using area of a circle formula.
* Calculated circumference of a circle and assigned it to a variable \_circum\_of\_circle\_ using circumference of a circle formula.
* Now, took the input of radius from the user.
* Calculated the area of circle again with the user input.

**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.

**Solution:**

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

y = x.split() #using split() to split the given sentence.

print("Sentence after splitting:", y)

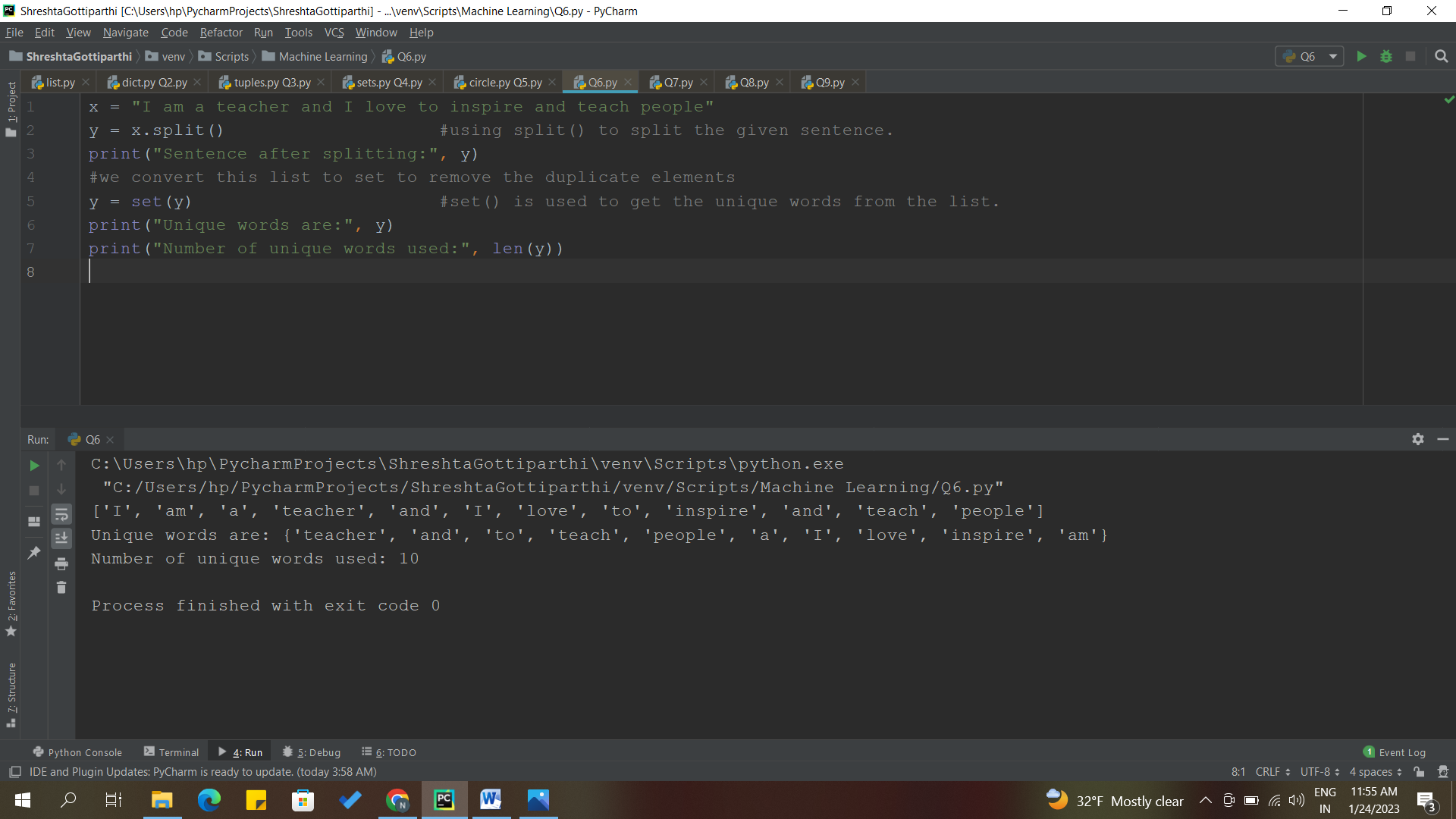
#we convert this list to set to remove the duplicate elements

y = set(y) #set() is used to get the unique words from the list.

print("Unique words are:", y)

print("Number of unique words used:", len(y))

**CODE AND OUTPUT:**



**DESCRIPTION:**

* Given a sentence, have stored it in a variable x.
* Using split() method, I have split the sentence, it converts to a list and stored it in y.
* We convert this list to set to remove the duplicate element using set(). set() is used to get the unique words from the list.
* For knowing no. of unique words used, used len() function.

**Question 7**

Use a tab escape sequence to get the following lines.

Name Age Country City

Asabeneh 250 Finland Helsinki

**Solution:**

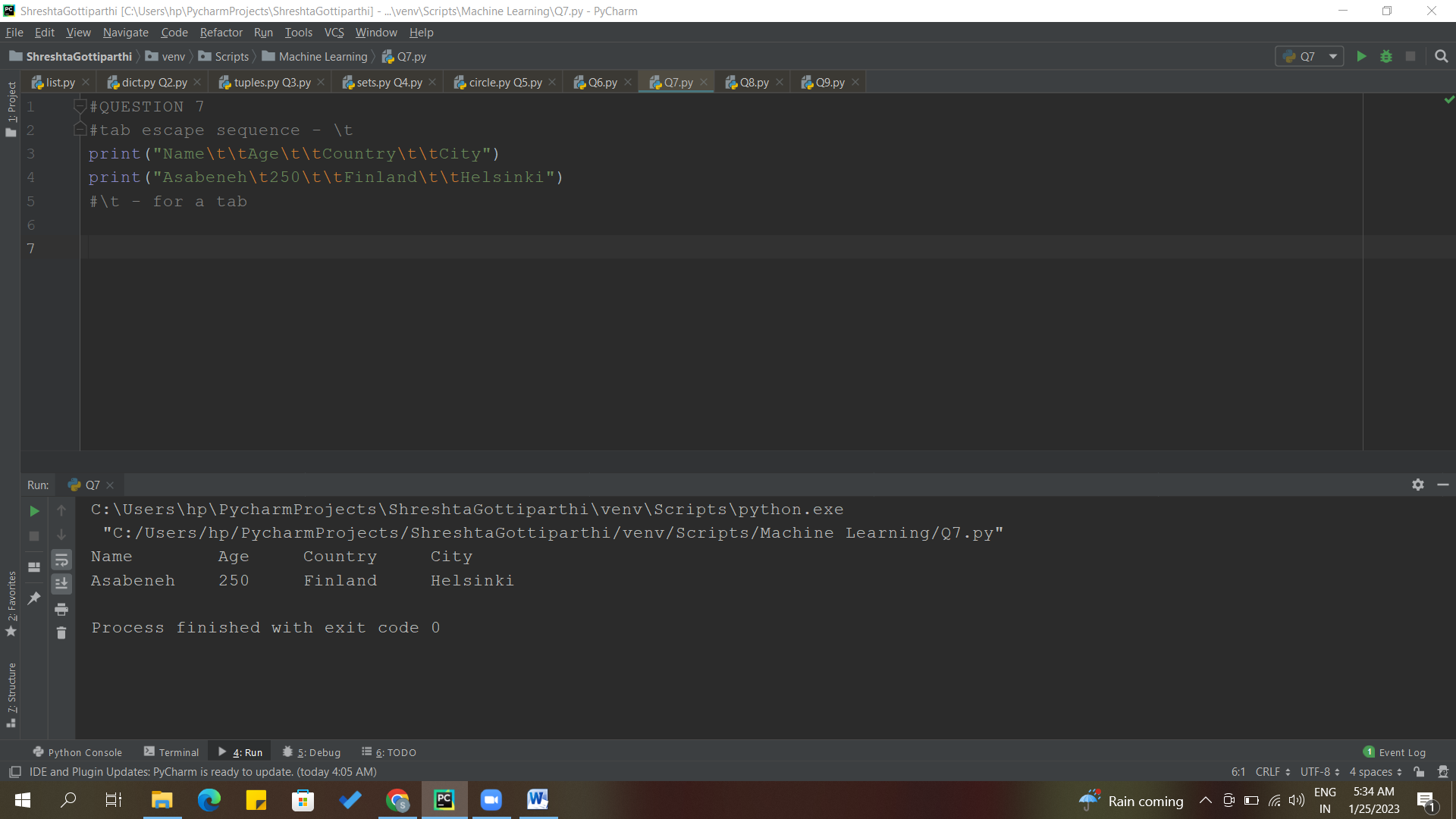
#tab escape sequence - \t

print("Name\t\tAge\t\tCountry\t\tCity")

print("Asabeneh\t250\t\tFinland\t\tHelsinki")

#\t - for a tab

**CODE AND OUTPUT:**

****

**DESCRIPTION:**

* Using tab escape sequence, we get the mentioned lines.
* I have used \t
* \t – for a tab

**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.”

**Solution:**

radius = 10

area = 3.14 \* radius \*\* 2

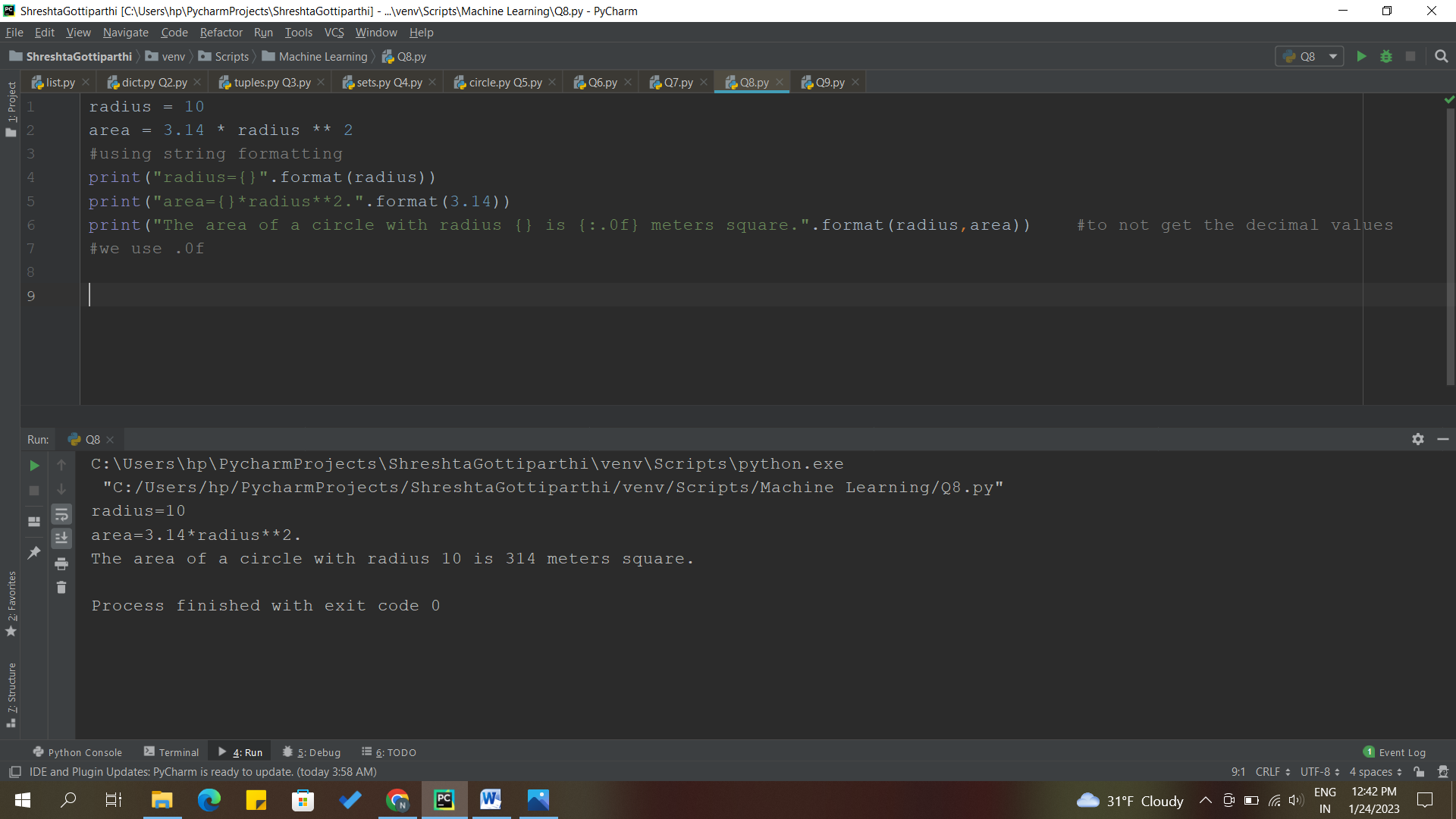
#using string formatting

print("radius={}".format(radius))

print("area={}\*radius\*\*2.".format(3.14))

print("The area of a circle with radius {} is {:.0f} meters square.".format(radius,area)) #to not get the decimal values, we use .0f

**CODE AND OUTPUT:**

****

**DESCRIPTION:**

* Using string formatting, displayed

radius = 10

area = 3.14 \* radius \*\* 2

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

* Used .format()

**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)

**Solution:**

N=int(input("Enter number of students:")) #taking the input from the user.

students\_weight\_lbs=[] #empty list of weigth in lbs.

while(N): #checking the condition by using while loop.

students\_weight\_lbs.append(int(input()))

N=N-1

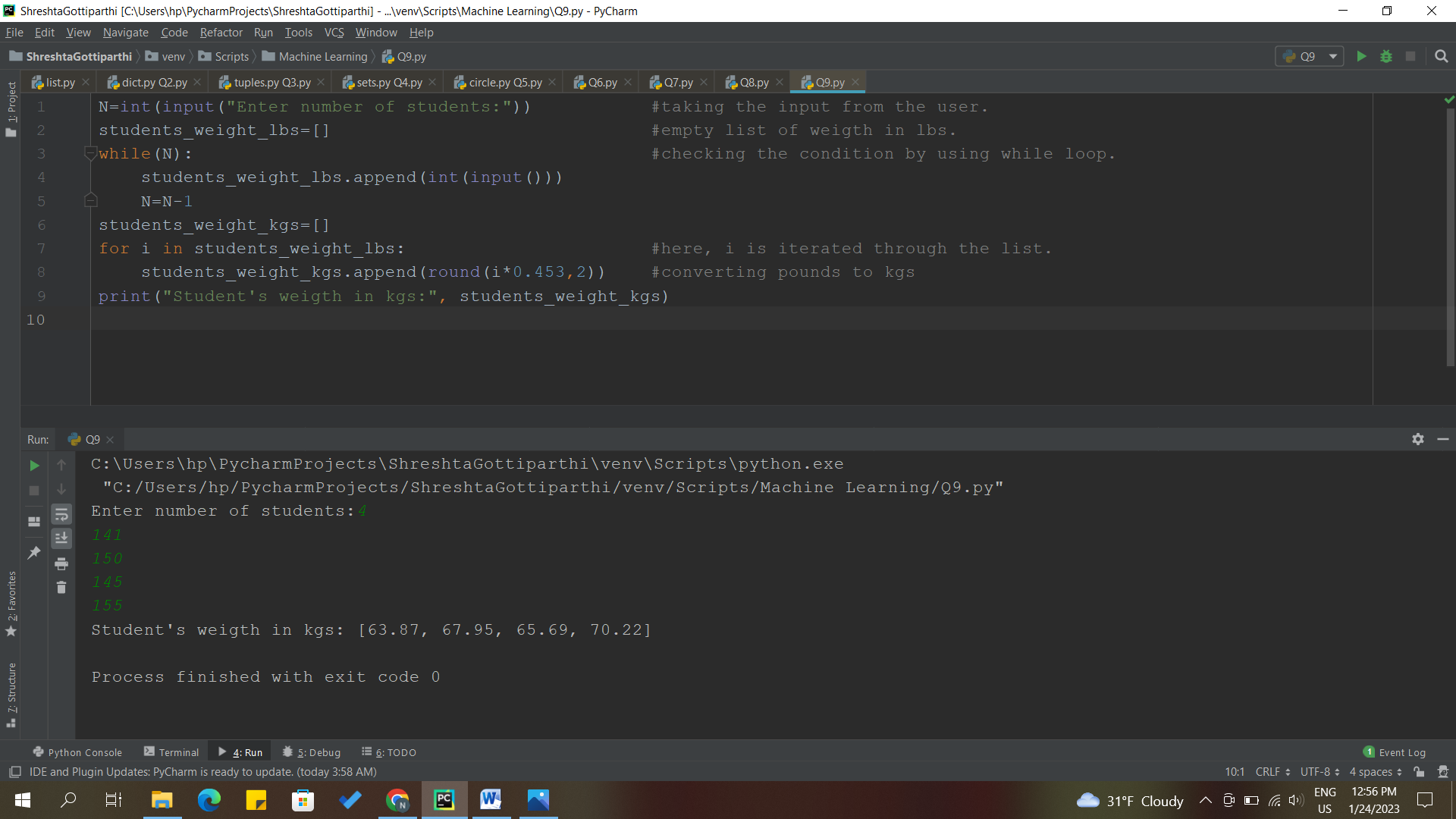
students\_weight\_kgs=[]

for i in students\_weight\_lbs: #here, i is iterated through the list.

students\_weight\_kgs.append(round(i\*0.453,2)) #converting pounds to kgs

print("Student's weigth in kgs:", students\_weight\_kgs)

**CODE AND OUTPUT:**



**DESCRIPTION:**

* N: No. of students, took it from the user.
* Took an empty list for the weights in lbs.
* Checking the condition using while loop. Appending the students weight in lbs with the number of students.
* Took an empty list for the weights in kgs.
* Used for loop for iterating through the list.
* Finally converted the weights in lbs to kgs.

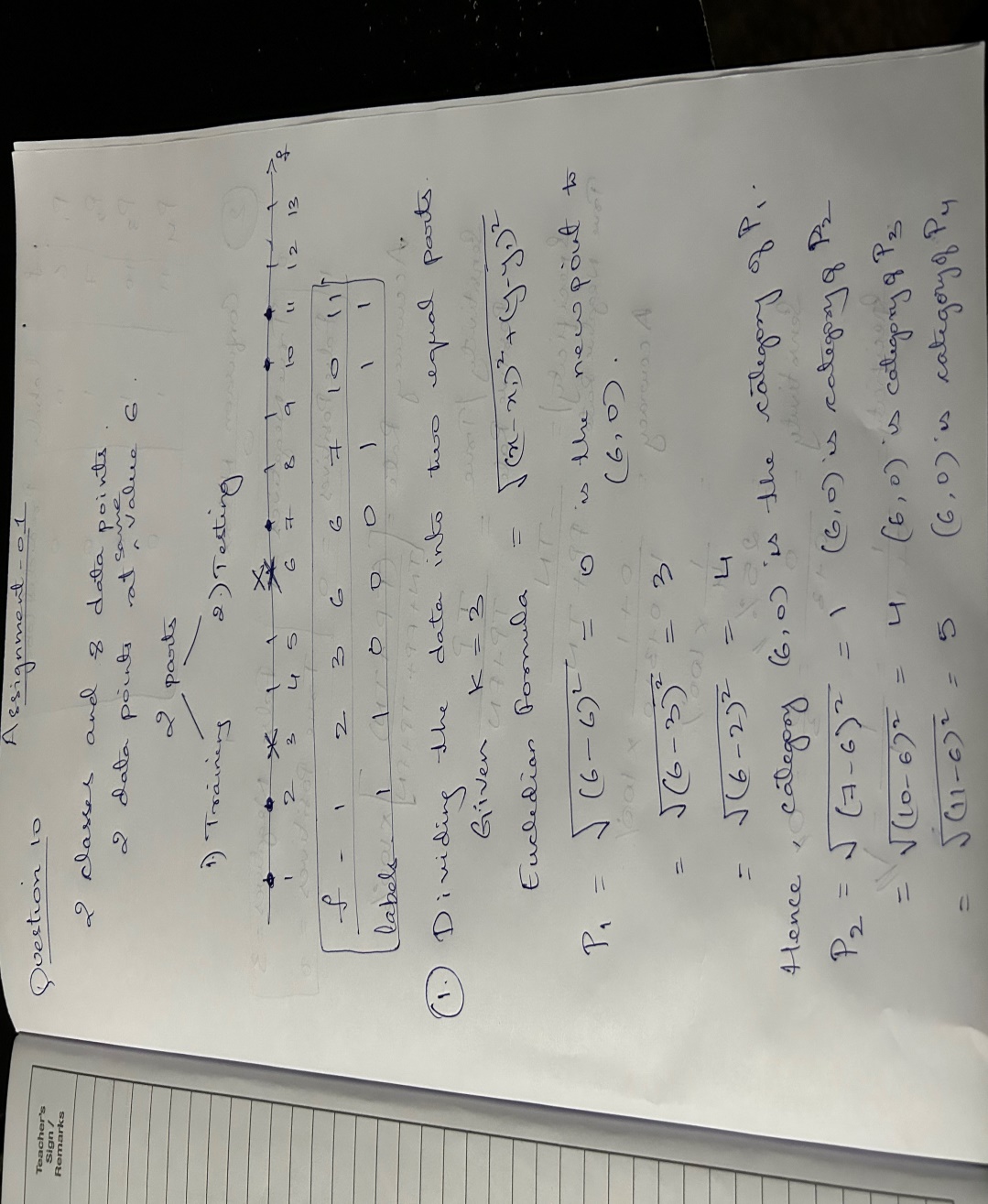
**Question 10**

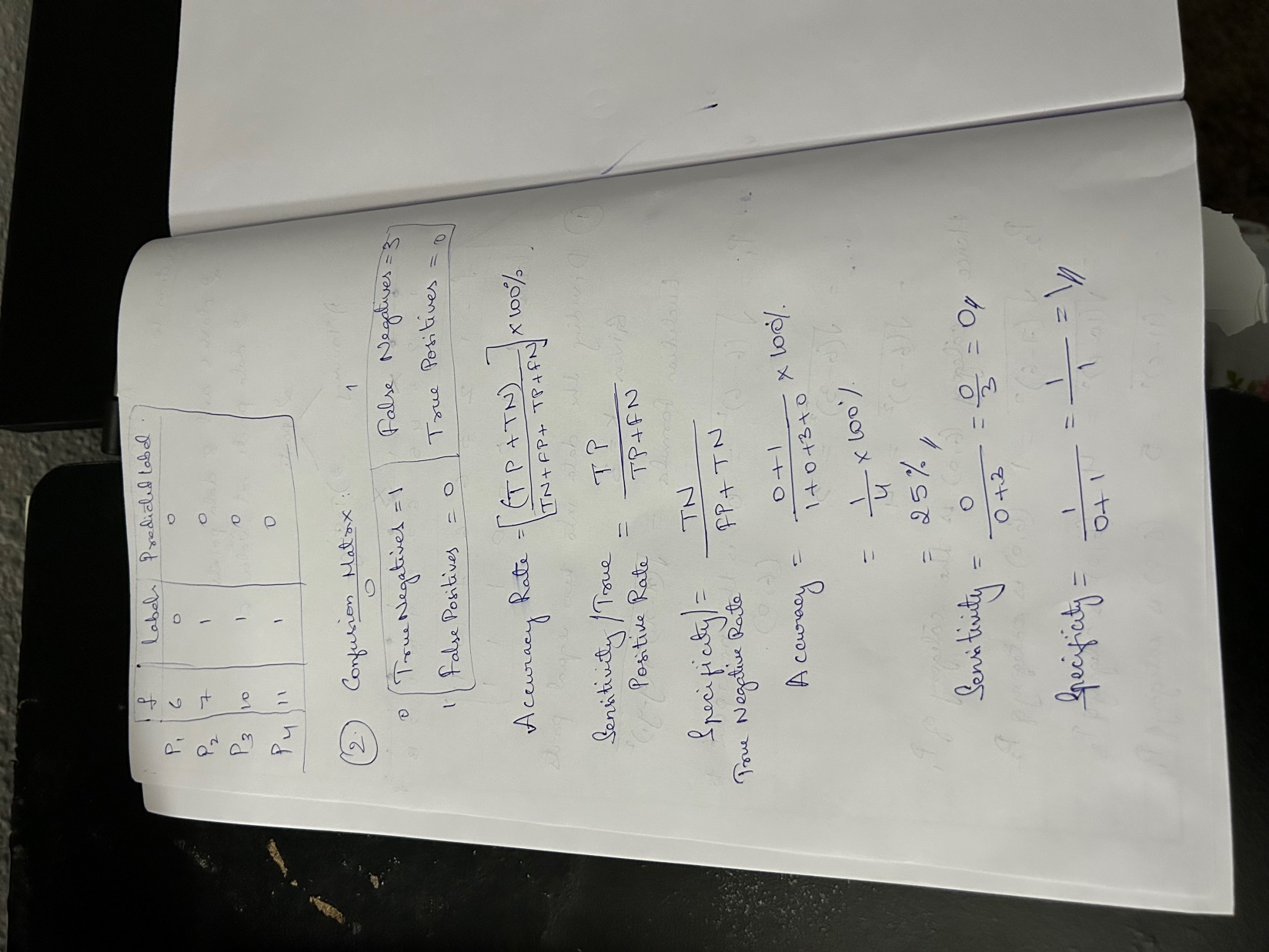
The diagram below shows a dataset with 2 classes and 8 data points, each with only one feature value, labeled f. Note that there are two data points with the same feature value of 6. These are shown as two x’s one above the other. Provide stepwise mathematical solution, do not write code for it.

1. 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.

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

**Solution:**

****

****