****

**Batch: B1 Roll No:1914078 Tutorial No.** :3

**Aim:** To implement Set data Structures in Python.



**Resources needed:** Python IDE



### Theory:

### Set is an unordered collection of unique items. Type of elements in a set need not be the same, and various mixed up data type values are passed. Set is defined by values separated by comma inside braces { }. No duplicates are allowed in sets. Even if an object is added more than once inside the curly brackets, only one copy is held in the set object. Hence sets does not support indexing, and hence no slicing. A set is mutable, but may not contain mutable items like a list, set, and dictionary. Order of elements in a set is undefined and is unchangeable.

### Syntax:

### set\_s={2,”hello”,3.4,”a”}

### set\_s1=set("Python")

### A set object has suitable methods to perform mathematical set operations like union, intersection, difference, etc. The inbuilt methods of sets include:

### add(): Adds a new element in the set object. Only one element at a time can be added.  Loops are used to add multiple elements at a time

### S1={"Python", "Java", "C++"}

### S1.add("Perl")

### update(): It accepts lists, strings, tuples as well as other sets as its arguments and add multiple items .

### S1.update(["C", "Basic"])

### discard(): Returns a set after removing an item from it. No changes are done if the item is not present.

### S1.discard(“Python")

### clear(): Removes the contents of the set object and results in an empty set.

### S1.clear()

### remove(): Returns a set after removing an item from it. Results in an error if the item is not present.

### S1.remove("C++")

### pop(): pops out the first element of the set and prints it. It cannot have arguments like index since indexing is not present.

### copy(): Creates a copy of the set object.

### Activity:

1. Use sets to check whether a sentence accepted from the user is a pangram(A pangram is a sentence containing every letter in the English Alphabet.)

**Code:**

user\_str = input("Enter a string to check for pangram : ")

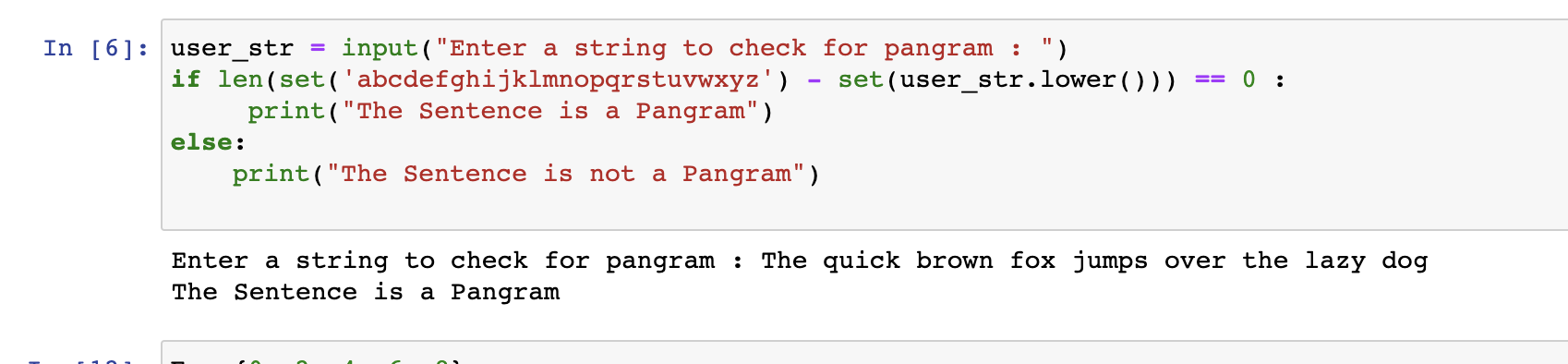
if len(set('abcdefghijklmnopqrstuvwxyz') - set(user\_str.lower())) == 0 :

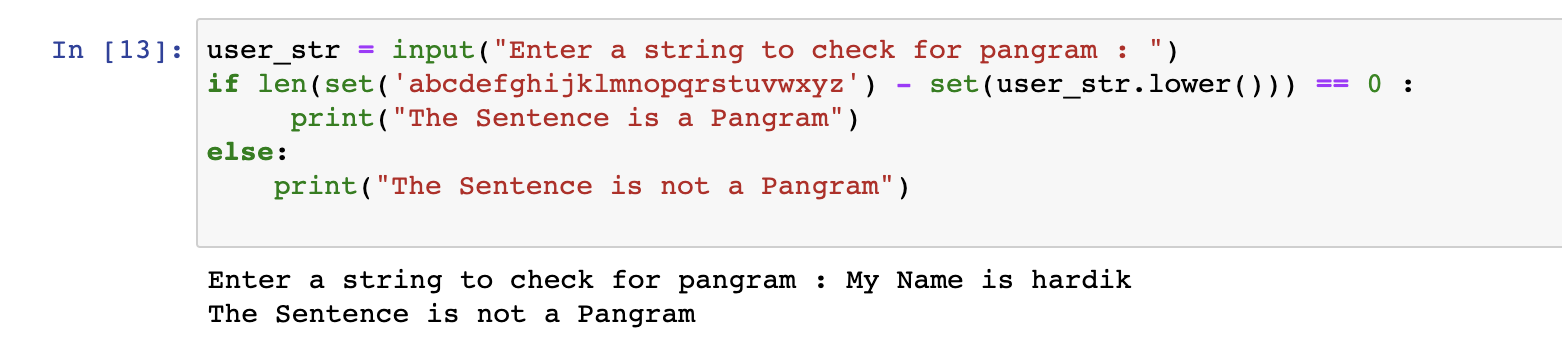
print("The Sentence is a Pangram")

else:

print("The Sentence is not a Pangram")

**Output:**

****

****

1. WAP to perform all set operations.

**Code:**

1. E = {0, 2, 4, 6, 8};

N = {1, 2, 3, 4, 5};

# set union

print("Union of E and N is: ",E | N)

# set intersection

print("Intersection of E and N is: ",E & N)

# set difference

print("Difference of E and N is: ",E - N)

# set symmetric difference

print("Symmetric difference of E and N is: ",E ^ N)

#add

E.add(10)

print("Set E after adding an element: ",E)

#update

N.update([6,7])

print("Set N after updating: ",N)

#discard

N.discard(7)

print("Set N after discarding 7 is: ",N)

#remove

N.remove(1)

print("Set N after removing 1 is: ",N)

#pop

E.pop()

print("Set E after popping an element: ",E)

#copy

X=E.copy()

print("X(copy of set E): ",X)

#clear

E.clear()

print("E after clearing: ",E)

2. engineers = {"John", "Jane", "Jack", "Janice"}

programmers = {"Jack", "Sam", "Susan", "Janice"}

l=len(engineers)

print("The length of engineers set is: ",l)

for name in [engineers,programmers]:

name.discard("Jack")

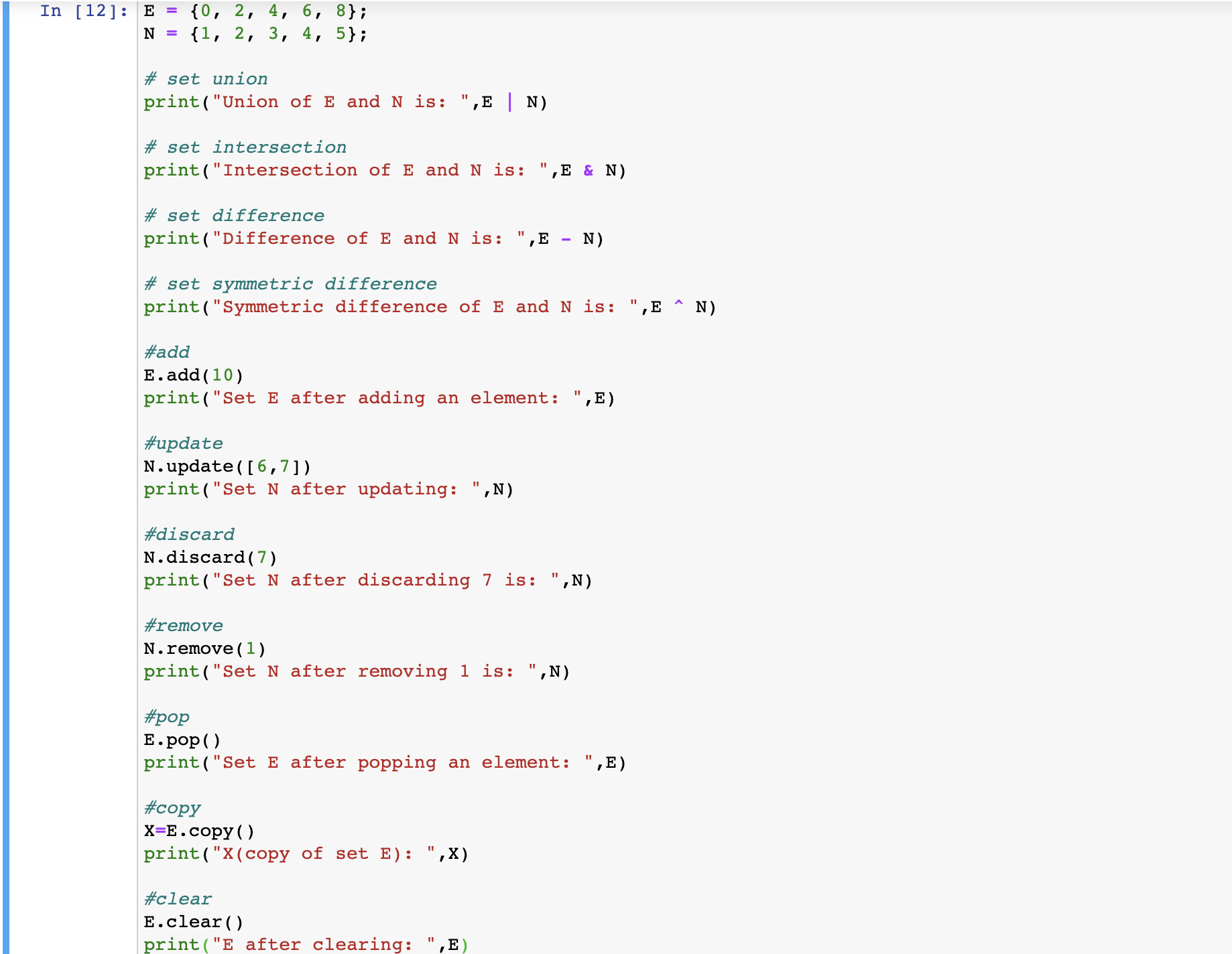
print(name)

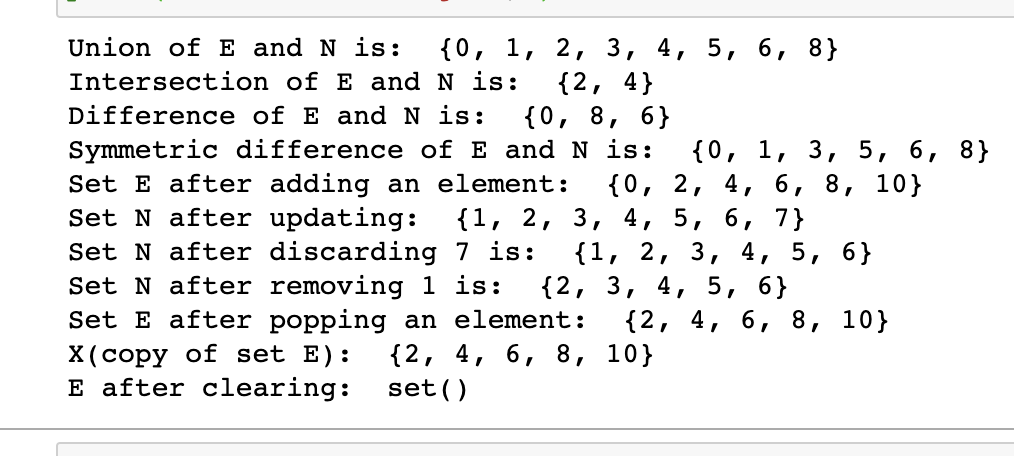
x="Sam"

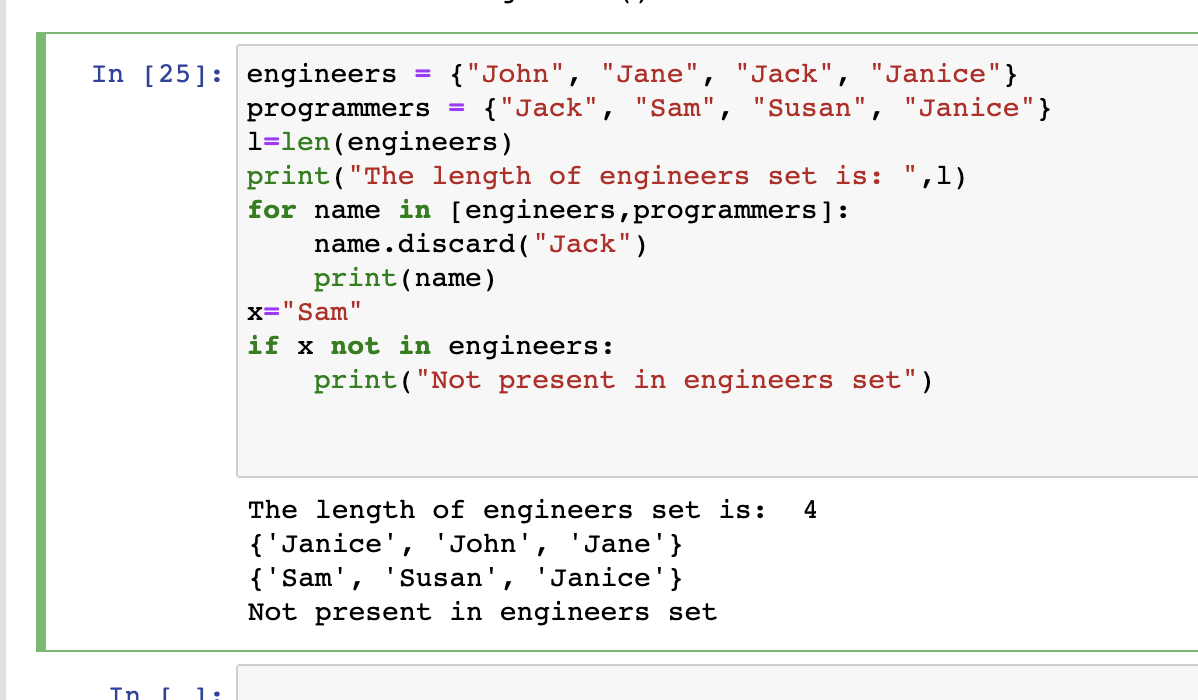
if x not in engineers:

print("Not present in engineers set")

**Output:**

****

****

****

### Post Tutorial Questions:

### Why are sets unordered?

**Ans:** As sets do not record element position or order of insertion,hence sets are unordered.This

is also why sets cannot have repeating values. Accordingly sets do not support indexing,

slicing, or other sequence-like behavior.That’s why sets do not contain mutable elements

such as lists or dictionaries.

### Outcomes: Use of Different Decision Making Statements and Functions in Python.

**Conclusion: (Conclusion to be based on the objectives and outcomes achieved)**

Through this experiment we illustrated the use of set data structures in python through use of

different decision making statements and functions such as add, remove, update, copy etc. and

also illustrated various set operations such as union etc.



**References:**

1. Reema Thareja , *Python Programing: Using Problem Solving Approach,* Oxford University Press, First Edition 2017, India
2. Dr. R. Nageswara Rao, *Core Python Programming,* Wiley Publication, Second Edition 2018,India
3. Sheetal Taneja and Naveen Kumar, *Python Programing: A Modular Approach,* Pearson India
4. Swarroop C.H, *Byte of python,* e-book, Kindle edition
5. Martin C Brown, *The Complete Reference Python,* Brandon A Nordin, First Edition 2001