**PRACTICAL-5**

**AIM:**

**a.** Write a program to create an empty set. Input the elements from user and write a for loop to add these elements onto the set.

**Source Code:**

my\_set = set()

num\_elements = int(input("Enter the number of elements to add to the set: "))

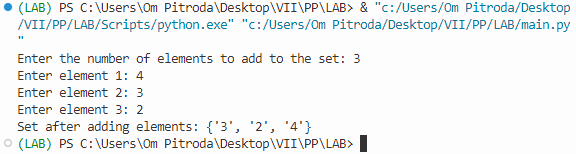
for i in range(num\_elements):

element = input(f"Enter element {i + 1}: ")

my\_set.add(element)

print("Set after adding elements:", my\_set)

**Output:**



**b.** Demonstrate the use of the following functions of Set Data Structure:

1. Operations on Set: difference(), difference\_update(), intersection(), intersection\_update(), symmetric\_difference(), symmetric\_difference\_update(), isdisjoint(), issuperset(), issubset()
2. Manipulating Set: discard(), add(), clear(), copy(), pop(), remove()

**­Source Code:**

set1 = {1, 2, 3, 4, 5}

set2 = {3, 4, 5, 6, 7}

diff\_set = set1.difference(set2)

set1.difference\_update(set2)

inter\_set = set1.intersection(set2)

set1.intersection\_update(set2)

sym\_diff\_set = set1.symmetric\_difference(set2)

set1.symmetric\_difference\_update(set2)

disjoint = set1.isdisjoint(set2)

superset = set1.issuperset(set2)

subset = set1.issubset(set2)

print("Difference of set1 and set2:", diff\_set)

print("Set1 after difference update:", set1)

print("Intersection of set1 and set2:", inter\_set)

print("Set1 after intersection update:", set1)

print("Symmetric difference of set1 and set2:", sym\_diff\_set)

print("Set1 after symmetric difference update:", set1)

print("Are set1 and set2 disjoint?", disjoint)

print("Is set1 a superset of set2?", superset)

print("Is set1 a subset of set2?", subset)

set1.discard(3)

set1.add(8)

set1.clear()

copy\_of\_set = set2.copy()

popped\_element = set2.pop()

set2.remove(5)

print("Set1 after discarding 3:", set1)

print("Set1 after adding 8:", set1)

print("Set1 after clearing:", set1)

print("Copy of Set2:", copy\_of\_set)

print("Popped Element from Set2:", popped\_element)

print("Set2 after removing 5:", set2)

**Output:**

