Shreya Nair-24BIT196

AIM: To implement various operations on sets in Python, including uppercase conversion, random number manipulation, and set modifications.

HARDWARE & SOFTWARE REQUIREMENTS: Hardware:16GB RAM, Intel Processor(i9), Software: Python (Version 3.x), Google Colab (Cloud-based)

SYSTEM CONFIGURATION: Operating System: Windows 11, IDE: Google Colab

THEORY: A set in Python is an unordered collection of unique elements. Sets do not allow duplicate values and support operations like union, intersection, and difference.

REFERENCES: Geeks for Geeks, Python Documentation: https://docs.python.org/3/

1. Write a program that converts words present in a list into uppercase and stores them in a set.

```
words=["python","semester","mentor","college"]
sets={word.upper() for word in words}
print(sets)
```

```
{'MENTOR', 'PYTHON', 'COLLEGE', 'SEMESTER'}
```

2. Write a program to create a set containing 10 random numbers in the range 15 to 45. Count how many of these numbers are less than 30. Delete all numbers that are greater than 35.

```
import random
no={random.randint(15,45) for _ in range(10)}
print("Random numbers:",no)
less=len([num for num in no if num<30])
print("Less than 30:",less)
no={num for num in no if num<=35}
print("After removing numbers greater than 35",no)
```

```
Random numbers: {32, 35, 36, 44, 17, 24, 26, 29, 30, 31}
Less than 30: 4
After removing numbers greater than 35 {32, 35, 17, 24, 26, 29, 30, 31}
```

3. Create an empty set. Write a program that adds five new names to this set, modifies one existing name and deletes two names from it.

```
name = set()
name.update(["Emily", "Barbara","Steve","David","Eve"])
print("Initial names:",name)
name.discard("CEve")
name.add("David")
name.discard("Steve")
name.discard("Emily")
print("Modified:", name)
```

```
Initial names: {'Emily', 'Eve', 'Steve', 'Barbara', 'David'}
Modified: {'Eve', 'Barbara', 'David'}
```

4. A set contains names which begin either with A or with B. Write a program to separate out the names into two sets, one containing names beginning with A and another with B.

```
name={"Amy","Bambi","Ayie","Bailey"}
A={nam for nam in name if nam.startswith("A")}
B={nam for nam in name if nam.startswith("B")}
print(A)
print(B)
```

```
{'Amy', 'Ayie'}
{'Bambi', 'Bailey'}
```