Name: Rajani Sankapal
Roll no: C-29

EXPERIMENT NO. 3

Q1. Create a dictionary with a student's name, age, and course. Add a new key grade, update age, delete course, and print all keys, values, and items.0

```
student = {
   "name": "Ram",
   "age": 20,
   "course": "Computer Science"
}
student["grade"] = "A"
student["age"] = 21
del student["course"]
print("Keys:", student.keys())
print("Values:", student.values())
print("Items:", student.items())
```

OUTPUT:-

```
Keys: dict_keys(['name', 'age', 'grade'])
Values: dict_values(['Ram', 21, 'A'])
Items: dict_items([('name', 'Ram'), ('age', 21), ('grade', 'A')])
```

2. Create a dictionary with 5 items. Add two new key-value pairs, update two existing values, delete one key using del and one using pop(), and iterate through keys and values.

```
data = {"a": 1, "b": 2, "c": 3, "d": 4, "e": 5}
data["f"] = 6
data["g"] = 7
data["b"] = 20
data["d"] = 40
del data["a"]
data.pop("c")
for key, value in data.items():
   print(key, ":", value)
```

OUTPUT:-

```
b : 20
d : 40
e : 5
f : 6
g : 7
```

3. Create a dictionary to store a book's title, author, price, and pages. Add a new key publisher, update the price, delete pages, check if author exists, and print the dictionary before and after each operation.

```
book = {"title": "Python Basics", "author": "John Doe", "price": 500, "pages": 350}
print("Original:", book)
book["publisher"] = "TechPress"
print("After adding publisher:", book)
book["price"] = 550
print("After updating price:", book)
del book["pages"]
print("After deleting pages:", book)
```

Name: Rajani Sankapal
Roll no: C-29

```
print("Author exists:", "author" in book)
```

OUTPUT:-

```
Original: {'title': 'Python Basics', 'author': 'John Doe', 'price': 500,
    'pages': 350}
After adding publisher: {'title': 'Python Basics', 'author': 'John Doe',
    'price': 500, 'pages': 350, 'publisher': 'TechPress'}
After updating price: {'title': 'Python Basics', 'author': 'John Doe',
    'price': 550, 'pages': 350, 'publisher': 'TechPress'}
After deleting pages: {'title': 'Python Basics', 'author': 'John Doe',
    'price': 550, 'publisher': 'TechPress'}
Author exists: True
```

4. Create a dictionary with employee details: id, name, department, and salary. Add a new key bonus, update salary, delete department, clear all items, and print dictionary after each operation.

```
employee = {"id": 101, "name": "Ram", "department": "IT", "salary": 40000}
print("Original:", employee)
employee["bonus"] = 5000
print("After adding bonus:", employee)
employee["salary"] = 45000
print("After updating salary:", employee)
del employee["department"]
print("After deleting department:", employee)
employee.clear()
print("After clearing all items:", employee)
```

OUTPUT:-

```
Original: {'id': 101, 'name': 'Ram', 'department': 'IT', 'salary': 40000}
After adding bonus: {'id': 101, 'name': 'Ram', 'department': 'IT', 'salary':
    40000, 'bonus': 5000}
After updating salary: {'id': 101, 'name': 'Ram', 'department': 'IT',
    'salary': 45000, 'bonus': 5000}
After deleting department: {'id': 101, 'name': 'Ram', 'salary': 45000,
    'bonus': 5000}
After clearing all items: {}
```

5. Create a dictionary of 4 fruits and their prices. Add a new fruit, update the price of an existing fruit, delete one fruit using del, pop another fruit, and print keys, values, and items.

```
fruits = {"apple": 100, "banana": 40, "mango": 80, "grapes": 60}
fruits["orange"] = 70
fruits["mango"] = 90
del fruits["banana"]
fruits.pop("grapes")
print("Keys:", fruits.keys())
print("Values:", fruits.values())
print("Items:", fruits.items())
```

OUTPUT:-

```
Keys: dict_keys(['apple', 'mango', 'orange'])
Values: dict_values([100, 90, 70])
Items: dict_items([('apple', 100), ('mango', 90), ('orange', 70)])
```

Name: Rajani Sankapal
Roll no: C-29

6. Create two sets of integers. Perform union, intersection, difference, symmetric difference, add a new element, remove an element, check subset and superset, and print the results after each operation.

```
A = {1, 2, 3, 4, 5}
B = {4, 5, 6, 7, 8}
print("Union:", A | B)
print("Intersection:", A & B)
print("Difference (A-B):", A - B)
print("Symmetric Difference:", A ^ B)
A.add(9)
B.remove(8)
print("After add/remove:", A, B)
print("A subset of B:", A.issubset(B))
print("A superset of B:", A.issuperset(B))
```

OUTPUT:-

```
Union: {1, 2, 3, 4, 5, 6, 7, 8}
Intersection: {4, 5}
Difference (A-B): {1, 2, 3}
Symmetric Difference: {1, 2, 3, 6, 7, 8}
After add/remove: {1, 2, 3, 4, 5, 9} {4, 5, 6, 7}
A subset of B: False
A superset of B: False
```

7. Create two sets of colors. Add a new color to each set, remove one color, find union, intersection, difference, symmetric difference, and check if one set is subset or superset of the other.

```
set1 = {"red", "blue", "green"}
set2 = {"yellow", "blue", "black"}
set1.add("white")
set2.add("pink")
set1.remove("red")
print("Union:", set1 | set2)
print("Intersection:", set1 & set2)
print("Difference:", set1 - set2)
print("Symmetric Difference:", set1 ^ set2)
print("Subset:", set1.issubset(set2))
print("Superset:", set1.issuperset(set2))
```

OUTPUT:-

```
Union: {'black', 'white', 'green', 'pink', 'yellow', 'blue'}
Intersection: {'blue'}
Difference: {'white', 'green'}
Symmetric Difference: {'pink', 'white', 'green', 'yellow', 'black'}
Subset: False
Superset: False
```

8. Create a set of 10 numbers. Add three new numbers, remove two numbers, print the set, and perform union, intersection, difference, and symmetric difference with another set.

```
nums1 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
nums2 = {5, 6, 7, 11, 12}
nums1.update([11, 12, 13])
```

Name: Rajani Sankapal
Roll no: C-29

```
nums1.remove(1)
nums1.discard(2)
print("Set after add/remove:", nums1)
print("Union:", nums1 | nums2)
print("Intersection:", nums1 & nums2)
print("Difference:", nums1 - nums2)
print("Symmetric Difference:", nums1 ^ nums2)
```

OUTPUT:-

```
Set after add/remove: {3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
Union: {3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
Intersection: {5, 6, 7, 11, 12}
Difference: {3, 4, 8, 9, 10, 13}
Symmetric Difference: {3, 4, 8, 9, 10, 13}
```

9. Create a dictionary with product details: name, quantity, price, and category. Add a discount key, update quantity and price, delete category, print keys, values, and items, and check if name exists.

```
product = {"name": "Laptop", "quantity": 10, "price": 50000, "category": "Electronics"}
product["discount"] = "10%"
product["quantity"] = 8
product["price"] = 48000
del product["category"]
print("Keys:", product.keys())
print("Values:", product.values())
print("Items:", product.items())
print("Name exists:", "name" in product)
```

OUTPUT:-

```
Keys: dict_keys(['name', 'quantity', 'price', 'discount'])
Values: dict_values(['Laptop', 8, 48000, '10%'])
Items: dict_items([('name', 'Laptop'), ('quantity', 8), ('price', 48000),
    ('discount', '10%')])
Name exists: True
```

10. Create a dictionary with 5 student names and their scores. Add scores for 2 more students, update 3 scores, delete 1 student using del and another using pop(), and iterate through keys and values.

```
students = {"Amit": 85, "Ram": 90, "Ravi": 78, "Sneha": 88, "Priya": 92}
students["Manav"] = 80
students["Isha"] = 95
students["Amit"] = 89
students["Aman"] = 93
students["Ravi"] = 82
del students["Sneha"]
students.pop("Priya")
for name, score in students.items():
  print(name, ":", score)
```

OUTPUT:-

Name: Rajani Sankapal
Roll no: C-29

```
Amit : 89
Ram : 90
Ravi : 82
Manav : 80
Isha : 95
Aman : 93
```

11. Create two sets of numbers representing exam scores. Perform union, intersection, difference, symmetric difference, add a new score, remove a score, and check if one set is subset or superset of the other.

```
setA = {75, 80, 85, 90}
setB = {85, 90, 95, 100}
print("Union:", setA | setB)
print("Intersection:", setA & setB)
print("Difference:", setA - setB)
print("Symmetric Difference:", setA ^ setB)
setA.add(70)
setB.remove(100)
print("After add/remove:", setA, setB)
print("Subset:", setA.issubset(setB))
print("Superset:", setA.issuperset(setB))
```

OUTPUT:-

```
Union: {100, 75, 80, 85, 90, 95}
Intersection: {90, 85}
Difference: {80, 75}
Symmetric Difference: {100, 75, 80, 95}
After add/remove: {70, 75, 80, 85, 90} {90, 85, 95}
Subset: False
Superset: False
```

12. Create a set of cities you have visited. Add 2 new cities, remove 1 city, perform union, intersection, difference, symmetric difference with another set of cities, and check subset/superset relationships.

```
cities1 = {"Mumbai", "Pune", "Delhi", "Goa"}
cities2 = {"Goa", "Bangalore", "Chennai"}
cities1.add("Hyderabad")
cities1.add("Kolkata")
cities1.remove("Delhi")
print("Union:", cities1 | cities2)
print("Intersection:", cities1 & cities2)
print("Difference:", cities1 - cities2)
print("Symmetric Difference:", cities1 ^ cities2)
print("Subset:", cities1.issubset(cities2))
print("Superset:", cities1.issuperset(cities2))
```

OUTPUT:-

Name: Rajani Sankapal
Roll no: C-29

```
Union: {'Hyderabad', 'Mumbai', 'Goa', 'Bangalore', 'Chennai', 'Pune',
    'Kolkata'}
Intersection: {'Goa'}
Difference: {'Hyderabad', 'Mumbai', 'Pune', 'Kolkata'}
Symmetric Difference: {'Bangalore', 'Hyderabad', 'Mumbai', 'Chennai', 'Pune'
    , 'Kolkata'}
Subset: False
Superset: False
```

13. Create a dictionary of 4 countries and their capitals. Add two more countries, update a capital, delete one country using del and another using pop(), print all keys, values, items, and check if a country exists.

```
countries = {"India": "Delhi", "USA": "Washington", "France": "Paris", "Japan": "Tokyo"}
countries["Germany"] = "Berlin"
countries["Italy"] = "Rome"
countries["France"] = "Marseille"
del countries["Japan"]
countries.pop("USA")
print("Keys:", countries.keys())
print("Values:", countries.values())
print("Items:", countries.items())
print("India exists:", "India" in countries)
```

OUTPUT:-

```
Keys: dict_keys(['India', 'France', 'Germany', 'Italy'])
Values: dict_values(['Delhi', 'Marseille', 'Berlin', 'Rome'])
Items: dict_items([('India', 'Delhi'), ('France', 'Marseille'), ('Germany',
    'Berlin'), ('Italy', 'Rome')])
India exists: True
```

14. Create a set of 8 favorite movies. Add 2 movies, remove 1 movie, perform union, intersection, difference, symmetric difference with another set, and check subset and superset.

```
movies1 = {"Inception", "Titanic", "Avatar", "Interstellar", "Joker", "Frozen", "Dune", "Up"}
movies2 = {"Avatar", "Frozen", "Barbie", "Dune"}
movies1.add("Oppenheimer")
movies1.add("Soul")
movies1.remove("Up")
print("Union:", movies1 | movies2)
print("Intersection:", movies1 & movies2)
print("Difference:", movies1 - movies2)
print("Symmetric Difference:", movies1 ^ movies2)
print("Subset:", movies1.issubset(movies2))
print("Superset:", movies1.issuperset(movies2))
```

OUTPUT:-

Name: Rajani Sankapal
Roll no: C-29

```
Union: {'Titanic', 'Dune', 'Joker', 'Oppenheimer', 'Soul', 'Inception',
    'Frozen', 'Interstellar', 'Barbie', 'Avatar'}
Intersection: {'Avatar', 'Dune', 'Frozen'}
Difference: {'Titanic', 'Joker', 'Oppenheimer', 'Soul', 'Inception',
    'Interstellar'}
Symmetric Difference: {'Titanic', 'Joker', 'Oppenheimer', 'Soul',
    'Inception', 'Interstellar', 'Barbie'}
Subset: False
Superset: False
```

15. Create a dictionary for a shopping list with items and quantities. Add 3 new items, update quantities for 2 items, delete 1 item using del and another using pop(), print all keys, values, items, and check if a specific item exists.

```
shopping = {"Milk": 2, "Bread": 1, "Eggs": 12, "Butter": 1}
shopping["Juice"] = 2
shopping["Apples"] = 6
shopping["Rice"] = 1
shopping["Milk"] = 3
shopping["Bread"] = 2
del shopping["Butter"]
shopping.pop("Eggs")
print("Keys:", shopping.keys())
print("Values:", shopping.values())
print("Items:", shopping.items())
print("Milk exists:", "Milk" in shopping)
```

OUTPUT:-

```
Keys: dict_keys(['Milk', 'Bread', 'Juice', 'Apples', 'Rice'])
Values: dict_values([3, 2, 2, 6, 1])
Items: dict_items([('Milk', 3), ('Bread', 2), ('Juice', 2), ('Apples', 6),
    ('Rice', 1)])
Milk exists: True
```