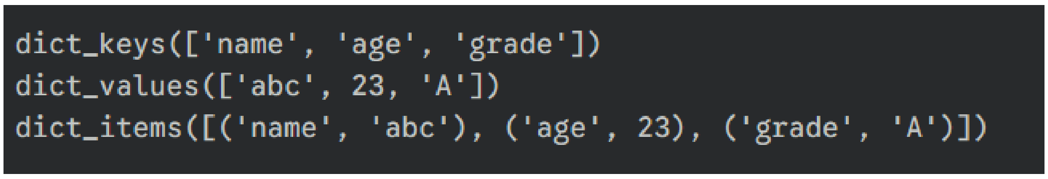
Name : Abhishek kadadevarmath
Roll no. : 46
Batch : C2

Assignment 3

1. 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.

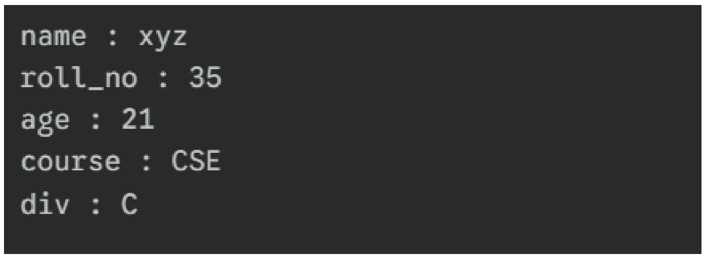
```
student = {"name": "abc", "age": 21, "course": "CSE"}
student["grade"] = "A"  # Add new key 'grade' [cite: 9, 10, 11]
student["age"] = 23    # Update 'age' [cite: 12]
del student["course"]  # Delete 'course' [cite: 13]
print(student.keys())
print(student.values())
give output for this
print(student.items())
```



```
dict_keys(['name', 'age', 'grade'])
dict_values(['abc', 23, 'A'])
dict_items([('name', 'abc'), ('age', 23), ('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.

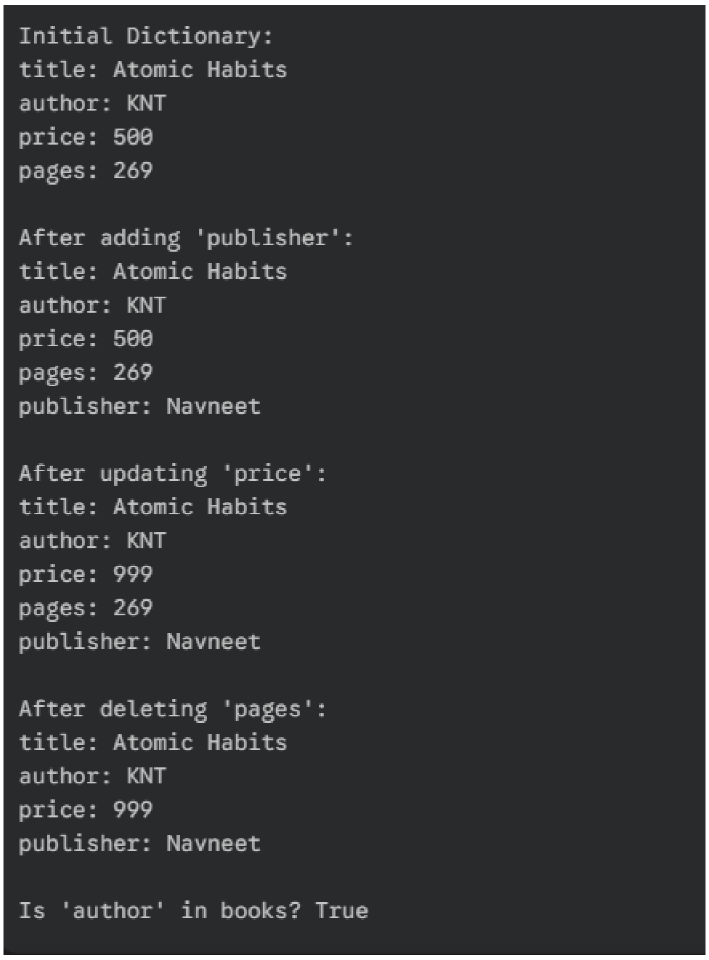
```
d = {"name": "abc", "class": "TY", "student": "YES", "roll_no": 35, "age": 21}
d["course"] = "CSE"  # Add 'course' [cite: 25]
d["div"] = "C"       # Add 'div' [cite: 26]
d["name"] = "xyz"  # Update 'name' [cite: 27]
d["class"] = "SY"   # Update 'class' [cite: 28]
del d["class"]       # Delete 'class' using del [cite: 29]
d.pop("student")     # Delete 'student' using pop() [cite: 30]
for key, value in d.items():
   print(key, ":", value)
```



```
name : xyz
roll_no : 35
age : 21
course : CSE
div : C
```

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.

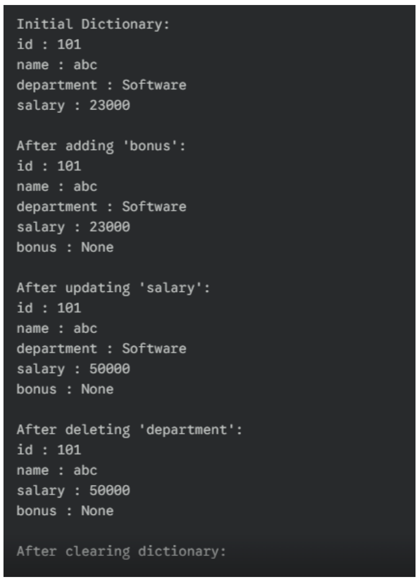
```
books = {"title": "Atomic Habits", "author": "KNT", "price": 500, "pages": 269}

# Initial Print [cite: 43]
print("Initial Dictionary:")
for key, value in books.items():
    print(key, ":", value)
print("\n")

# Add 'publisher' [cite: 46]
books["publisher"] = "Navneet"
print("After adding 'publisher':")
for key, value in books.items():
    print(key, ":", value)
print("\n")

# Update 'price' [cite: 49]
books["price"] = 999
print("After updating 'price':")
for key, value in books.items():
    print(key, ":", value)
print("\n")

# Delete 'pages' [cite: 53]
del books["pages"]
print("After deleting 'pages':")
for key, value in books.items():
    print(key, ":", value)
print("\n")

# Check if 'author' exists [cite: 57]
print("Is 'author' in books?", "author" in books)
```


```
Initial Dictionary:
title: Atomic Habits
author: KNT
price: 500
pages: 269

After adding 'publisher':
title: Atomic Habits
author: KNT
price: 500
pages: 269
publisher: Navneet

After updating 'price':
title: Atomic Habits
author: KNT
price: 999
pages: 269
publisher: Navneet

After deleting 'pages':
title: Atomic Habits
author: KNT
price: 999
publisher: Navneet

Is 'author' in books? True
```

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.

```
emp = {"id": 101, "name": "abc", "department": "Software", "salary": 23000}


# Initial Print [cite: 85, 86]
print("Initial Dictionary:")
for key, value in emp.items():
    print(key, ":", value)
print("\n")


# Add 'bonus' [cite: 88]
emp["bonus"] = None
print("After adding 'bonus':")
for key, value in emp.items():
    print(key, ":", value)
print("\n")


# Update 'salary' [cite: 91]
```

```
emp["salary"] = 50000
print("After updating 'salary':")
for key, value in emp.items():
    print(key, ":", value)
print("\n")

# Delete 'department' [cite: 95]
del emp["department"]
print("After deleting 'department':")
for key, value in emp.items():
    print(key, ":", value)
print("\n")

# Clear all items [cite: 99]
emp.clear()
print("After clearing dictionary:")
for key, value in emp.items():
    print(key, ":", value)
print("\n")
```

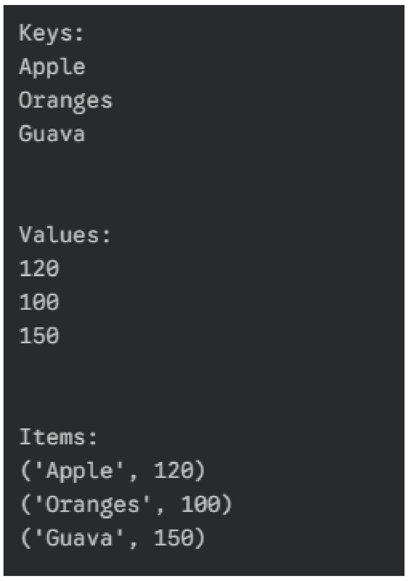


```
Initial Dictionary:
id : 101
name : abc
department : Software
salary : 23000

After adding 'bonus':
id : 101
name : abc
department : Software
salary : 23000
bonus : None

After updating 'salary':
id : 101
name : abc
department : Software
salary : 50000
bonus : None

After deleting 'department':
id : 101
name : abc
salary : 50000
bonus : None

After clearing dictionary:
```

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.

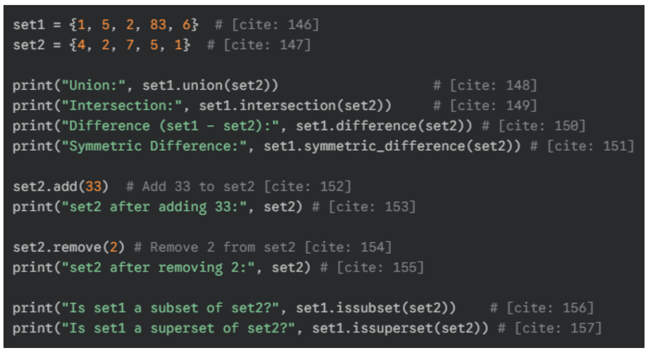
```
fruits = {"Apple": 230, "Banana": 60, "Cherry": 150, "Oranges": 100}
```

```
fruits["Guava"] = 150   # Add 'Guava' [cite: 122]
fruits["Apple"] = 120  # Update 'Apple' price [cite: 123]
del fruits["Banana"]   # Delete 'Banana' using del [cite: 124]
fruits.pop("Cherry")   # Delete 'Cherry' using pop() [cite: 125]

print("Keys:")
for key in fruits.keys():
   print(key)
print("\n")

print("Values:")
for value in fruits.values():
   print(value)
print("\n")

print("Items:")
for item in fruits.items():
   print(item)
print("\n")
```

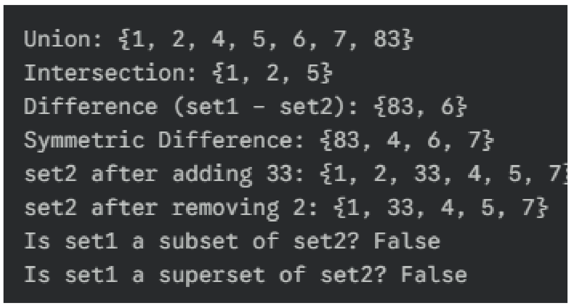


```
Keys:
Apple
Oranges
Guava


Values:
120
100
150


Items:
('Apple', 120)
('Oranges', 100)
('Guava', 150)
```

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.

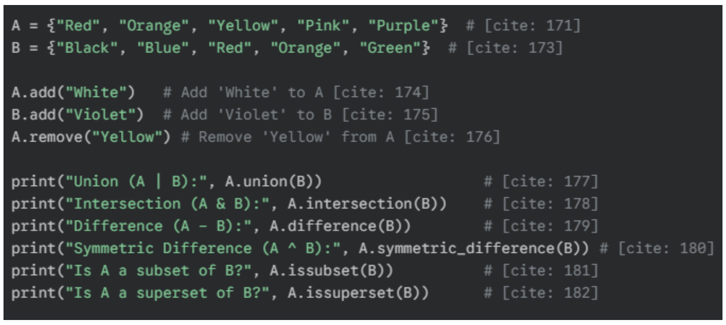


```python
set1 = {1, 5, 2, 83, 6}  # [cite: 146]
set2 = {4, 2, 7, 5, 1}  # [cite: 147]

print("Union:", set1.union(set2))                   # [cite: 148]
print("Intersection:", set1.intersection(set2))     # [cite: 149]
print("Difference (set1 - set2):", set1.difference(set2)) # [cite: 150]
print("Symmetric Difference:", set1.symmetric_difference(set2)) # [cite: 151]

set2.add(33)  # Add 33 to set2 [cite: 152]
print("set2 after adding 33:", set2) # [cite: 153]

set2.remove(2) # Remove 2 from set2 [cite: 154]
print("set2 after removing 2:", set2) # [cite: 155]

print("Is set1 a subset of set2?", set1.issubset(set2))    # [cite: 156]
print("Is set1 a superset of set2?", set1.issuperset(set2)) # [cite: 157]
```

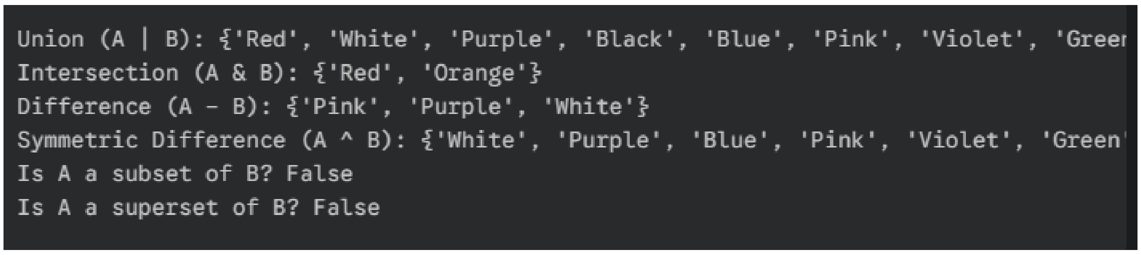


```
Union: {1, 2, 4, 5, 6, 7, 83}
Intersection: {1, 2, 5}
Difference (set1 - set2): {83, 6}
Symmetric Difference: {83, 4, 6, 7}
set2 after adding 33: {1, 2, 33, 4, 5, 7
set2 after removing 2: {1, 33, 4, 5, 7}
Is set1 a subset of set2? False
Is set1 a superset of set2? False
```

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.

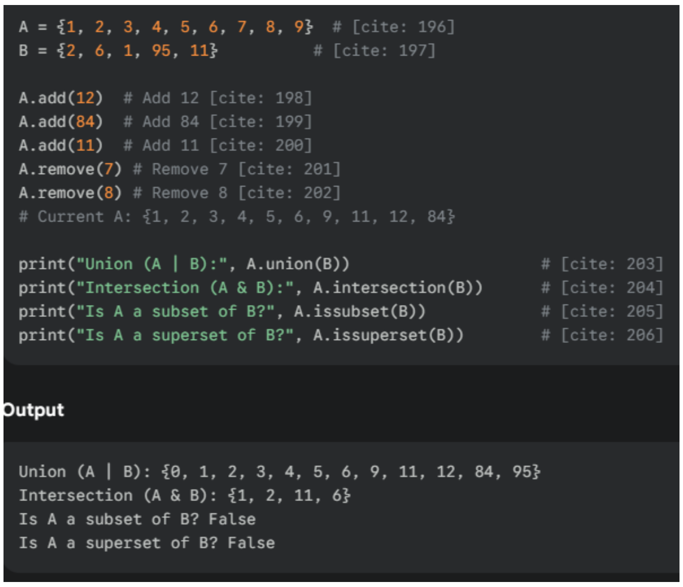


```python
A = {"Red", "Orange", "Yellow", "Pink", "Purple"}  # [cite: 171]
B = {"Black", "Blue", "Red", "Orange", "Green"}  # [cite: 173]

A.add("White")   # Add 'White' to A [cite: 174]
B.add("Violet")  # Add 'Violet' to B [cite: 175]
A.remove("Yellow") # Remove 'Yellow' from A [cite: 176]

print("Union (A | B):", A.union(B))                 # [cite: 177]
print("Intersection (A & B):", A.intersection(B))   # [cite: 178]
print("Difference (A - B):", A.difference(B))       # [cite: 179]
print("Symmetric Difference (A ^ B):", A.symmetric_difference(B)) # [cite: 180]
print("Is A a subset of B?", A.issubset(B))         # [cite: 181]
print("Is A a superset of B?", A.issuperset(B))     # [cite: 182]
```



```
Union (A | B): {'Red', 'White', 'Purple', 'Black', 'Blue', 'Pink', 'Violet', 'Gree
Intersection (A & B): {'Red', 'Orange'}
Difference (A - B): {'Pink', 'Purple', 'White'}
Symmetric Difference (A ^ B): {'White', 'Purple', 'Blue', 'Pink', 'Violet', 'Green'
Is A a subset of B? False
Is A a superset of B? 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.

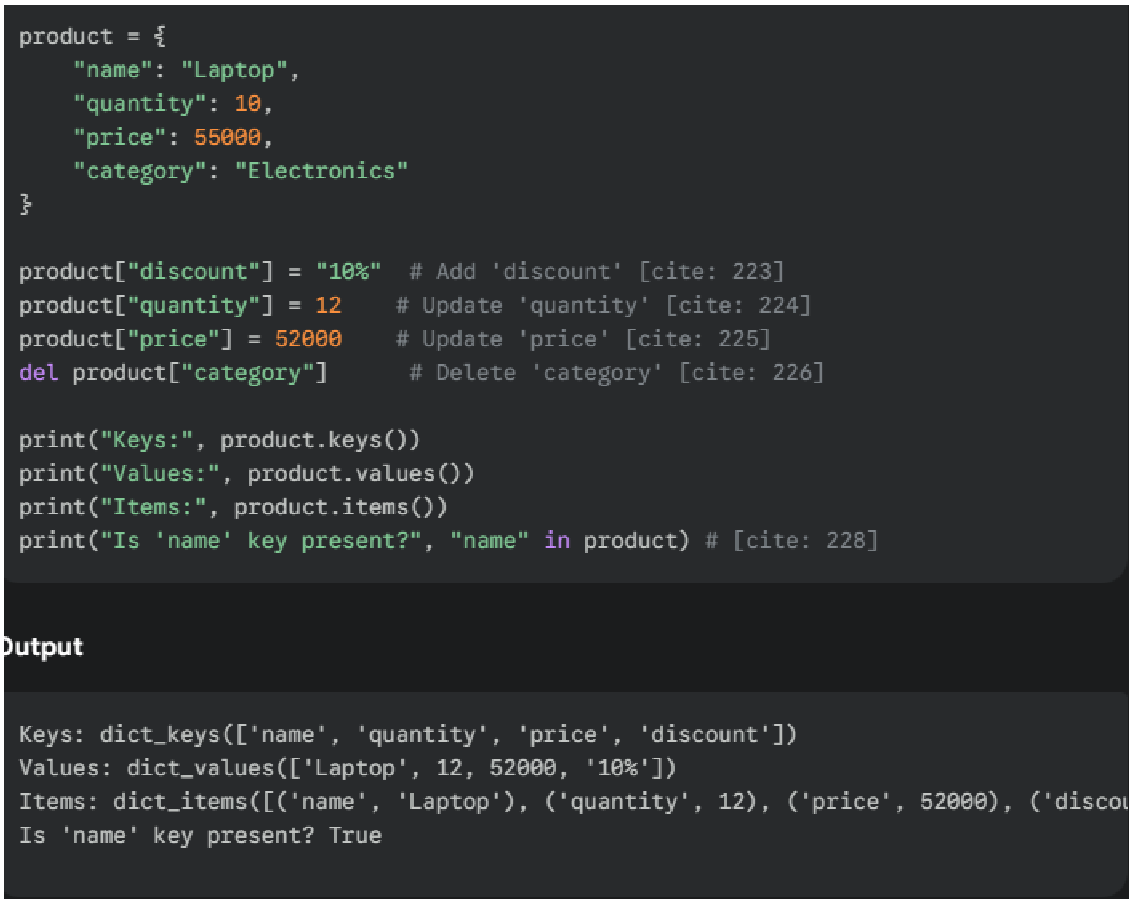


```python
A = {1, 2, 3, 4, 5, 6, 7, 8, 9}  # [cite: 196]
B = {2, 6, 1, 95, 11}          # [cite: 197]

A.add(12)  # Add 12 [cite: 198]
A.add(84)  # Add 84 [cite: 199]
A.add(11)  # Add 11 [cite: 200]
A.remove(7) # Remove 7 [cite: 201]
A.remove(8) # Remove 8 [cite: 202]
# Current A: {1, 2, 3, 4, 5, 6, 9, 11, 12, 84}

print("Union (A | B):", A.union(B))                    # [cite: 203]
print("Intersection (A & B):", A.intersection(B))      # [cite: 204]
print("Is A a subset of B?", A.issubset(B))            # [cite: 205]
print("Is A a superset of B?", A.issuperset(B))        # [cite: 206]
```

Output

```
Union (A | B): {0, 1, 2, 3, 4, 5, 6, 9, 11, 12, 84, 95}
Intersection (A & B): {1, 2, 11, 6}
Is A a subset of B? False
Is A a superset of B? False
```

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.

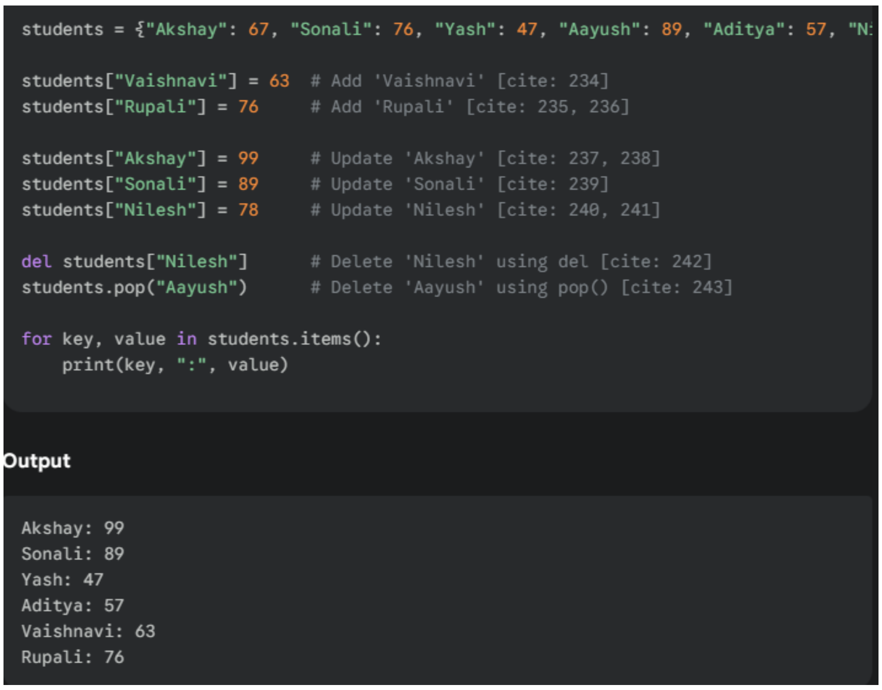


```
product = {
    "name": "Laptop",
    "quantity": 10,
    "price": 55000,
    "category": "Electronics"
}

product["discount"] = "10%"  # Add 'discount' [cite: 223]
product["quantity"] = 12    # Update 'quantity' [cite: 224]
product["price"] = 52000    # Update 'price' [cite: 225]
del product["category"]      # Delete 'category' [cite: 226]

print("Keys:", product.keys())
print("Values:", product.values())
print("Items:", product.items())
print("Is 'name' key present?", "name" in product) # [cite: 228]
```

Dutput

```
Keys: dict_keys(['name', 'quantity', 'price', 'discount'])
Values: dict_values(['Laptop', 12, 52000, '10%'])
Items: dict_items([('name', 'Laptop'), ('quantity', 12), ('price', 52000), ('discou
Is 'name' key present? 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.

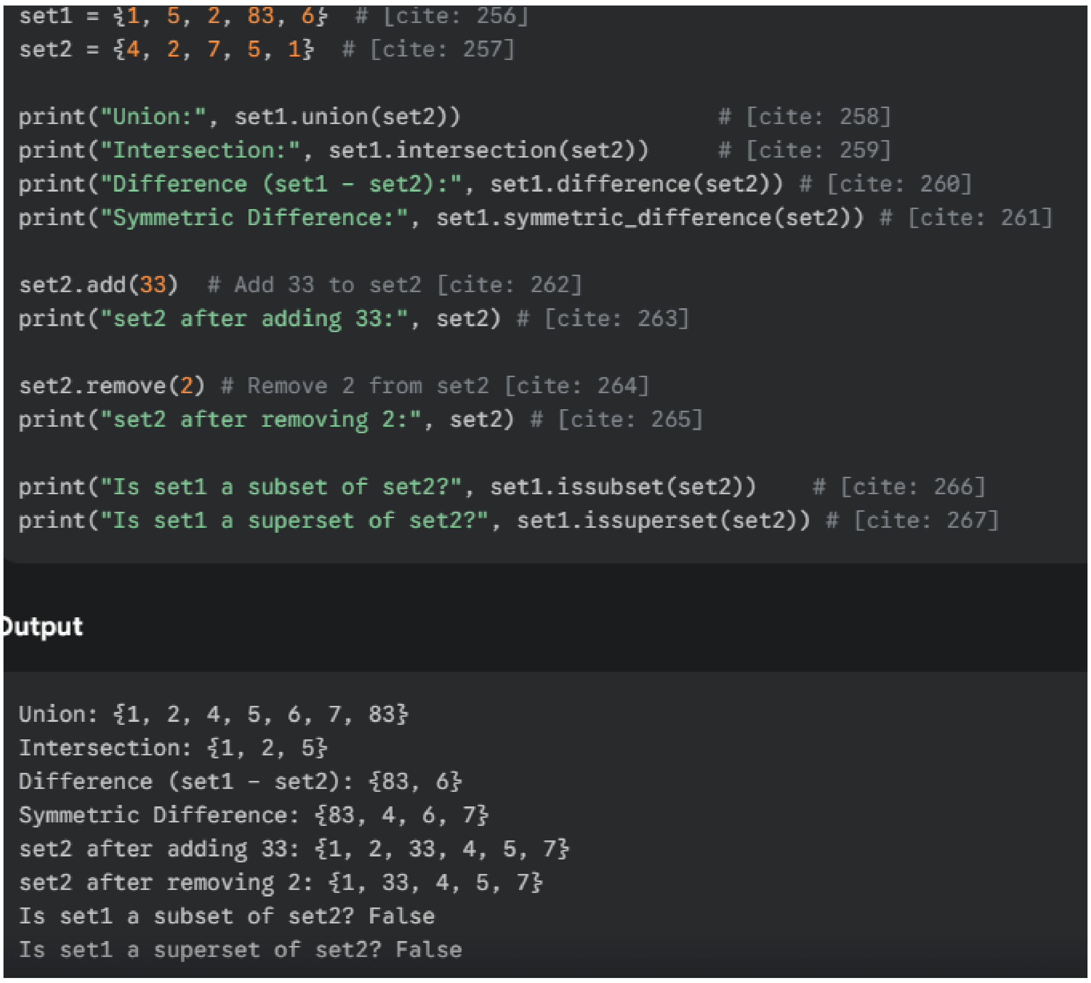


```
students = {"Akshay": 67, "Sonali": 76, "Yash": 47, "Aayush": 89, "Aditya": 57, "N

students["Vaishnavi"] = 63  # Add 'Vaishnavi' [cite: 234]
students["Rupali"] = 76     # Add 'Rupali' [cite: 235, 236]

students["Akshay"] = 99     # Update 'Akshay' [cite: 237, 238]
students["Sonali"] = 89     # Update 'Sonali' [cite: 239]
students["Nilesh"] = 78     # Update 'Nilesh' [cite: 240, 241]

del students["Nilesh"]      # Delete 'Nilesh' using del [cite: 242]
students.pop("Aayush")      # Delete 'Aayush' using pop() [cite: 243]

for key, value in students.items():
    print(key, ":", value)
```

Output

```
Akshay: 99
Sonali: 89
Yash: 47
Aditya: 57
Vaishnavi: 63
Rupali: 76
```

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.



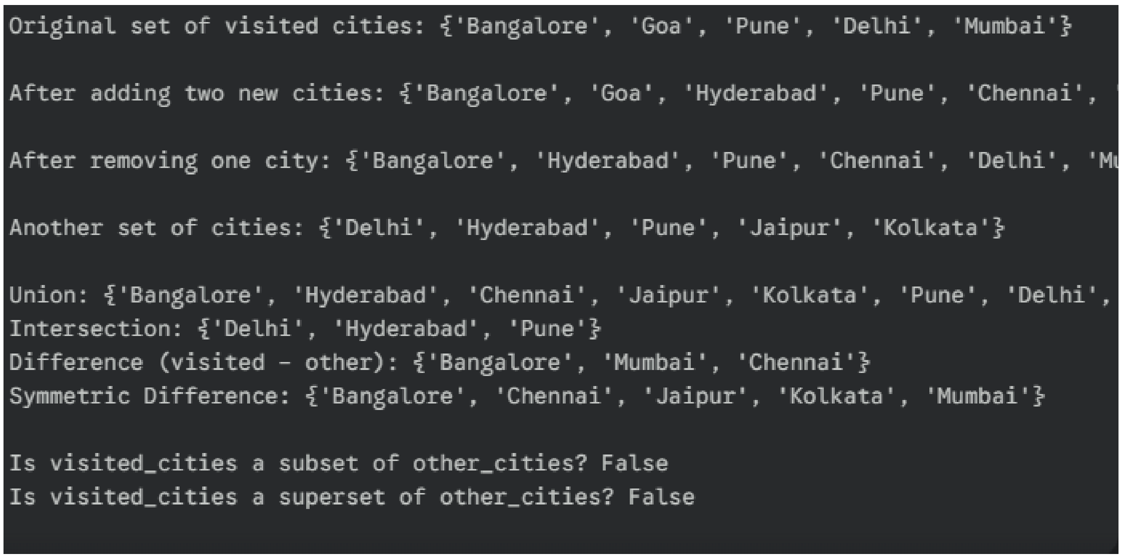
```
set1 = {1, 5, 2, 83, 6}  # [cite: 256]
set2 = {4, 2, 7, 5, 1}  # [cite: 257]

print("Union:", set1.union(set2))                    # [cite: 258]
print("Intersection:", set1.intersection(set2))      # [cite: 259]
print("Difference (set1 - set2):", set1.difference(set2)) # [cite: 260]
print("Symmetric Difference:", set1.symmetric_difference(set2)) # [cite: 261]

set2.add(33)  # Add 33 to set2 [cite: 262]
print("set2 after adding 33:", set2) # [cite: 263]

set2.remove(2) # Remove 2 from set2 [cite: 264]
print("set2 after removing 2:", set2) # [cite: 265]

print("Is set1 a subset of set2?", set1.issubset(set2))    # [cite: 266]
print("Is set1 a superset of set2?", set1.issuperset(set2)) # [cite: 267]
```

Output

```
Union: {1, 2, 4, 5, 6, 7, 83}
Intersection: {1, 2, 5}
Difference (set1 - set2): {83, 6}
Symmetric Difference: {83, 4, 6, 7}
set2 after adding 33: {1, 2, 33, 4, 5, 7}
set2 after removing 2: {1, 33, 4, 5, 7}
Is set1 a subset of set2? False
Is set1 a superset of set2? False
```

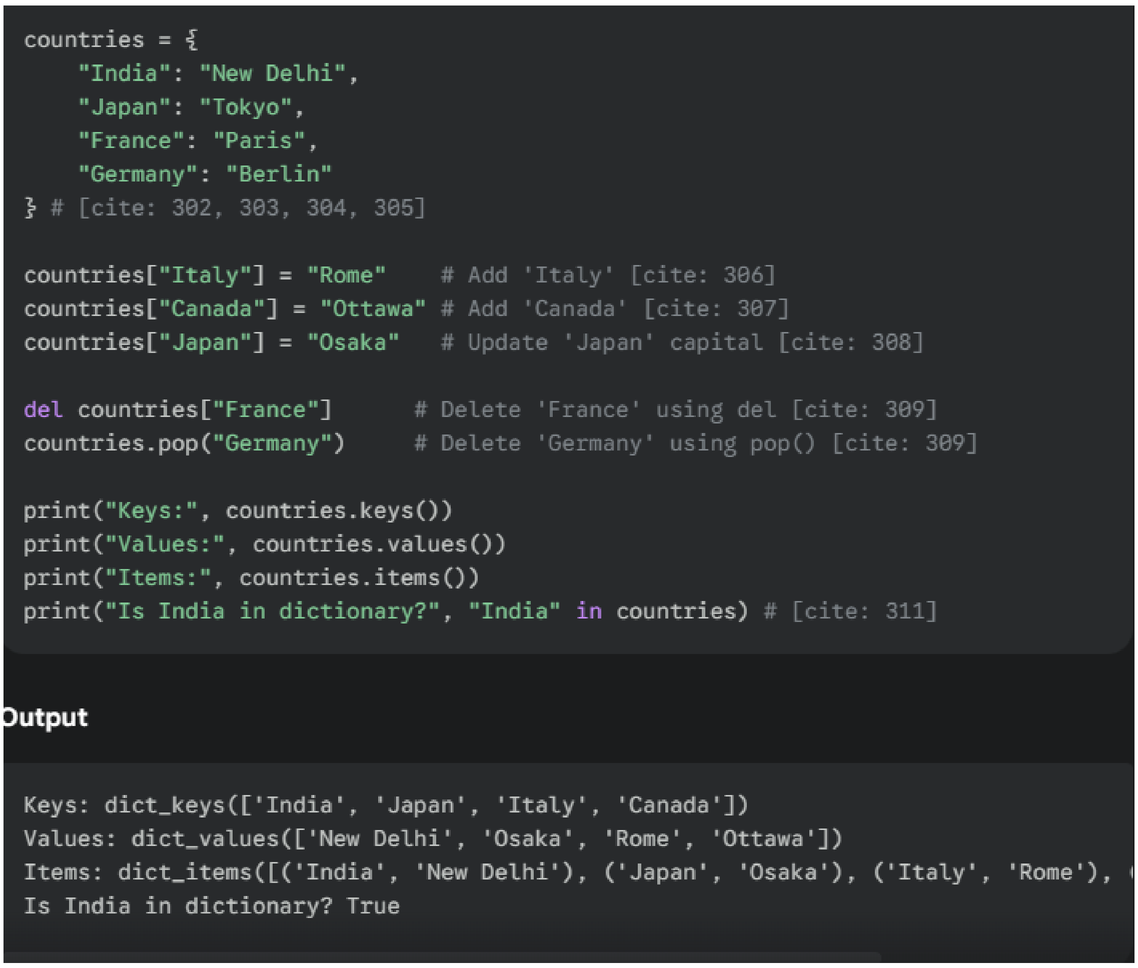
12.Create a set of cities you have visited. Add 2 new cities, remove 1 city, perform union, intersection, difference, symmetric difference with another set of cities, and check subset/superset relationships.

```
visited_cities = {"Mumbai", "Pune", "Goa", "Delhi", "Bangalore"} # [cite: 278]
print("Original set of visited cities:", visited_cities)

visited_cities.add("Hyderabad")  # Add 'Hyderabad' [cite: 279]
visited_cities.add("Chennai")    # Add 'Chennai' [cite: 280]
print("\nAfter adding two new cities:", visited_cities) # [cite: 281]

visited_cities.remove("Goa")     # Remove 'Goa' [cite: 281]
print("\nAfter removing one city:", visited_cities) # [cite: 282]

other_cities = {"Kolkata", "Delhi", "Pune", "Jaipur", "Hyderabad"} # [cite: 283]
print("\nAnother set of cities:", other_cities)

print("\nUnion:", visited_cities.union(other_cities)) # [cite: 284]
print("Intersection:", visited_cities.intersection(other_cities)) # [cite: 285]
print("Difference (visited - other):", visited_cities.difference(other_cities)) # [cite: 285]
print("Symmetric Difference:", visited_cities.symmetric_difference(other_cities)) # [cite: 286]
print("\nIs visited_cities a subset of other_cities?", visited_cities.issubset(other_cities)) # [cite: 286]
print("Is visited_cities a superset of other_cities?", visited_cities.issuperset(other_cities)) # [cite: 286]
```



```
Original set of visited cities: {'Bangalore', 'Goa', 'Pune', 'Delhi', 'Mumbai'}

After adding two new cities: {'Bangalore', 'Goa', 'Hyderabad', 'Pune', 'Chennai', 

After removing one city: {'Bangalore', 'Hyderabad', 'Pune', 'Chennai', 'Delhi', 'M

Another set of cities: {'Delhi', 'Hyderabad', 'Pune', 'Jaipur', 'Kolkata'}

Union: {'Bangalore', 'Hyderabad', 'Chennai', 'Jaipur', 'Kolkata', 'Pune', 'Delhi',
Intersection: {'Delhi', 'Hyderabad', 'Pune'}
Difference (visited - other): {'Bangalore', 'Mumbai', 'Chennai'}
Symmetric Difference: {'Bangalore', 'Chennai', 'Jaipur', 'Kolkata', 'Mumbai'}

Is visited_cities a subset of other_cities? False
Is visited_cities a superset of other_cities? 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.



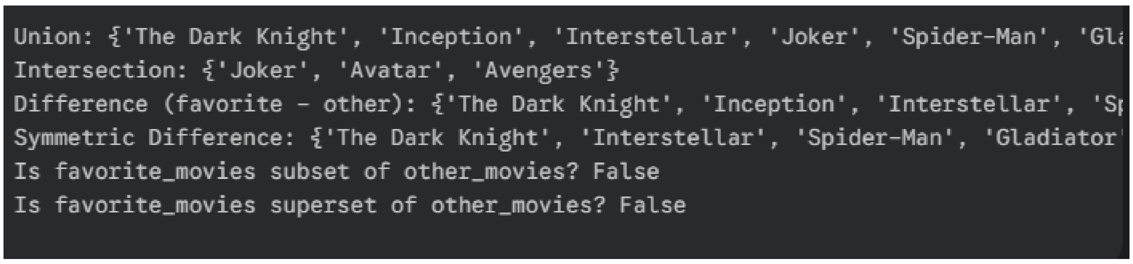
```
countries = {
    "India": "New Delhi",
    "Japan": "Tokyo",
    "France": "Paris",
    "Germany": "Berlin"
} # [cite: 302, 303, 304, 305]

countries["Italy"] = "Rome"    # Add 'Italy' [cite: 306]
countries["Canada"] = "Ottawa" # Add 'Canada' [cite: 307]
countries["Japan"] = "Osaka"   # Update 'Japan' capital [cite: 308]

del countries["France"]      # Delete 'France' using del [cite: 309]
countries.pop("Germany")     # Delete 'Germany' using pop() [cite: 309]

print("Keys:", countries.keys())
print("Values:", countries.values())
print("Items:", countries.items())
print("Is India in dictionary?", "India" in countries) # [cite: 311]
```

Output

```
Keys: dict_keys(['India', 'Japan', 'Italy', 'Canada'])
Values: dict_values(['New Delhi', 'Osaka', 'Rome', 'Ottawa'])
Items: dict_items([('India', 'New Delhi'), ('Japan', 'Osaka'), ('Italy', 'Rome'), (
Is India in dictionary? True
```

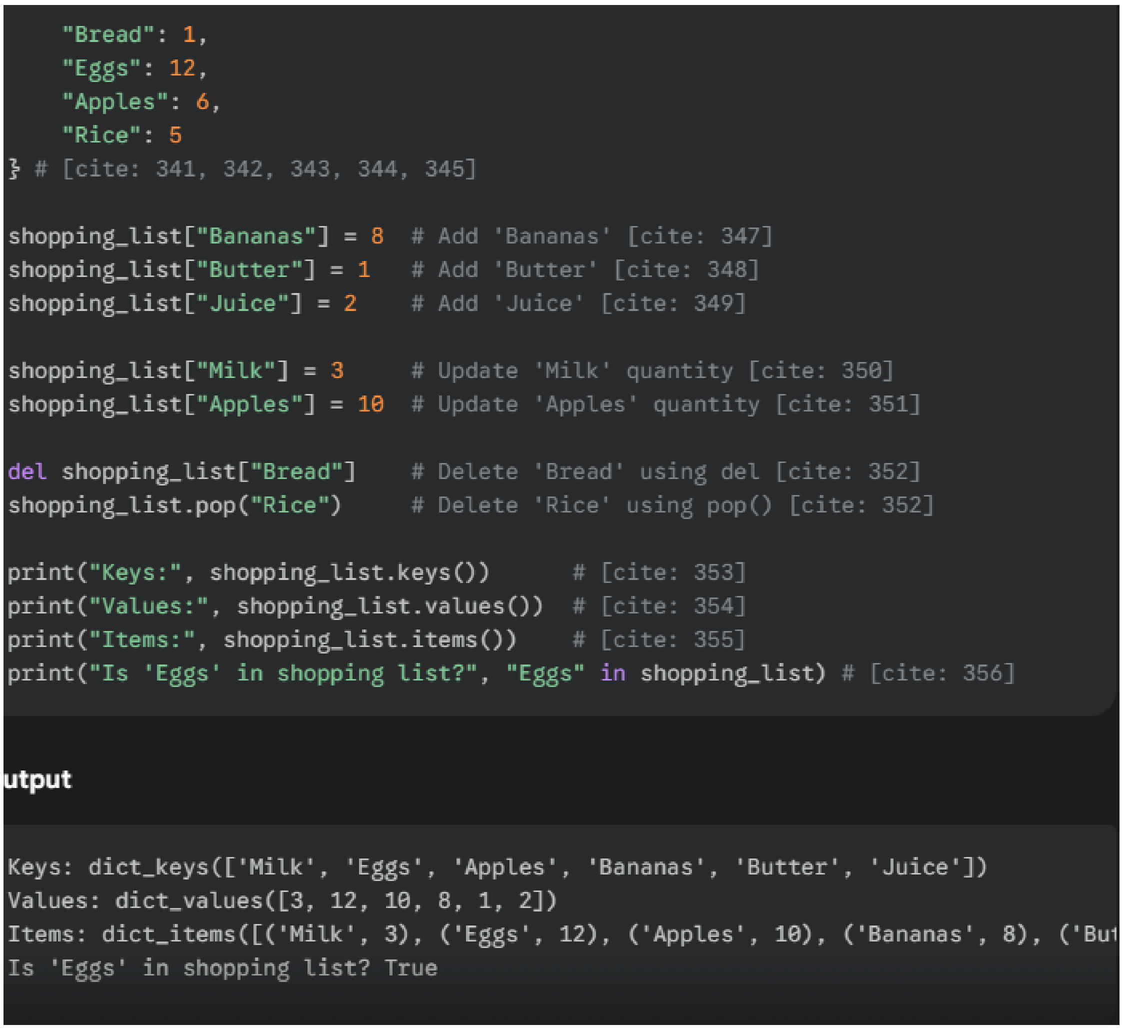
14.Create a set of 8 favourite movies. Add 2 movies, remove 1 movie, perform union, intersection, difference, symmetric difference with another set, and check subset and superset.

```
    favorite_movies = {"Inception", "Interstellar", "Avatar", "Titanic", "The Dark Knight", "Joker",
"Gladiator", "Avengers"} # [cite: 318]

favorite_movies.add("Spider-Man") # Add 'Spider-Man' [cite: 319]
favorite_movies.add("Iron Man")   # Add 'Iron Man' [cite: 320]
favorite_movies.remove("Titanic") # Remove 'Titanic' [cite: 321]

other_movies = {"Avatar", "Joker", "Frozen", "Moana", "Avengers"} # [cite: 322]

print("Union:", favorite_movies.union(other_movies)) # [cite: 323]
print("Intersection:", favorite_movies.intersection(other_movies)) # [cite: 324]
print("Difference (favorite - other):", favorite_movies.difference(other_movies)) # [cite: 325]
print("Symmetric Difference:", favorite_movies.symmetric_difference(other_movies)) # [cite: 326]
print("Is favorite_movies subset of other_movies?", favorite_movies.issubset(other_movies)) # [cite:
327]
print("Is favorite_movies superset of other_movies?", favorite_movies.issuperset(other_movies)) # [cite:
328]
```



```
Union: {'The Dark Knight', 'Inception', 'Interstellar', 'Joker', 'Spider-Man', 'Gla
Intersection: {'Joker', 'Avatar', 'Avengers'}
Difference (favorite - other): {'The Dark Knight', 'Inception', 'Interstellar', 'Sp
Symmetric Difference: {'The Dark Knight', 'Interstellar', 'Spider-Man', 'Gladiator'
Is favorite_movies subset of other_movies? False
Is favorite_movies superset of other_movies? 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.



```python
    "Bread": 1,
    "Eggs": 12,
    "Apples": 6,
    "Rice": 5
} # [cite: 341, 342, 343, 344, 345]

shopping_list["Bananas"] = 8  # Add 'Bananas' [cite: 347]
shopping_list["Butter"] = 1   # Add 'Butter' [cite: 348]
shopping_list["Juice"] = 2    # Add 'Juice' [cite: 349]

shopping_list["Milk"] = 3     # Update 'Milk' quantity [cite: 350]
shopping_list["Apples"] = 10  # Update 'Apples' quantity [cite: 351]

del shopping_list["Bread"]    # Delete 'Bread' using del [cite: 352]
shopping_list.pop("Rice")     # Delete 'Rice' using pop() [cite: 352]

print("Keys:", shopping_list.keys())      # [cite: 353]
print("Values:", shopping_list.values())  # [cite: 354]
print("Items:", shopping_list.items())    # [cite: 355]
print("Is 'Eggs' in shopping list?", "Eggs" in shopping_list) # [cite: 356]
```

utput

```
Keys: dict_keys(['Milk', 'Eggs', 'Apples', 'Bananas', 'Butter', 'Juice'])
Values: dict_values([3, 12, 10, 8, 1, 2])
Items: dict_items([('Milk', 3), ('Eggs', 12), ('Apples', 10), ('Bananas', 8), ('But
Is 'Eggs' in shopping list? True
```