PYTHON DATA STRUCTURES TASKS

 # LIST TASKS

 # Create a list of 10 student names
students = ["Alice", "Bob", "Charlie", "David", "Eva", "Frank", "Grace", "Hannah", "Ian", "Julia"]

 # Print the 3rd and 7th name
print("3rd student:", students[2]) # Index 2 = 3rd element
print("7th student:", students[6]) # Index 6 = 7th element

 # Add a new name and remove one old name
students.append("Kevin") # Adding new name
students.remove("Charlie") # Removing "Charlie"
print("Updated student list:", students)

 # Find the length of the list
print("Length of list:", len(students))

 # Write a program to find the sum of all even numbers in a list
numbers = [1, 2, 3, 4, 5, 6, 10, 15, 20]
even_sum = sum([num for num in numbers if num % 2 == 0])
print("Sum of even numbers:", even_sum)

 # Slice the list to get only the first 5 names
print("First 5 students:", students[:5])

 # Iterate through the list using a for loop and print each element
print("All students:")
for name in students:
  print(name)

---

[1]
0s

```
# 1️⃣ LIST TASKS

# Create a list of 10 student names
students = ["Alice", "Bob", "Charlie", "David", "Eva", "Frank", "Grace", "Hannah", "Ian", "Julia"]

# Print the 3rd and 7th name
print("3rd student:", students[2])   # Index 2 = 3rd element
print("7th student:", students[6])   # Index 6 = 7th element

# Add a new name and remove one old name
students.append("Kevin")   # Adding new name
students.remove("Charlie") # Removing "Charlie"
print("Updated student list:", students)

# Find the length of the list
print("Length of list:", len(students))

# Write a program to find the sum of all even numbers in a list
numbers = [1, 2, 3, 4, 5, 6, 10, 15, 20]
even_sum = sum([num for num in numbers if num % 2 == 0])
print("Sum of even numbers:", even_sum)

# Slice the list to get only the first 5 names
print("First 5 students:", students[:5])

# Iterate through the list using a for loop and print each element
print("All students:")
for name in students:
    print(name)
```

```
3rd student: Charlie
7th student: Grace
Updated student list: ['Alice', 'Bob', 'David', 'Eva', 'Frank', 'Grace', 'Hannah', 'Ian', 'Julia', 'Kevin']
Length of list: 10
Sum of even numbers: 42
First 5 students: ['Alice', 'Bob', 'David', 'Eva', 'Frank']
All students:
Alice
Bob
David
Eva
Frank
Grace
Hannah
Ian
Julia
Kevin
```

 # TUPLE TASKS

 # Create a tuple of 5 cities
cities = ("New York", "Paris", "Tokyo", "London", "Paris")

 # Print the first and last city
print("First city:", cities[0])
print("Last city:", cities[-1])

 # Try to change one city (will throw an error if uncommented)
 # cities[1] = "Berlin" # ❌ Tuples are immutable

 # Count how many times a particular city appears
print("Count of 'Paris':", cities.count("Paris"))

 # Find the index of a given city
print("Index of 'Tokyo':", cities.index("Tokyo"))

 # Iterate and print cities in uppercase
print("Cities in uppercase:")
for city in cities:
    print(city.upper())

---

[3]
✓ 0s

```
# 2️⃣ TUPLE TASKS

# Create a tuple of 5 cities
cities = ("New York", "Paris", "Tokyo", "London", "Paris")

# Print the first and last city
print("First city:", cities[0])
print("Last city:", cities[-1])

# Try to change one city (will throw an error if uncommented)
# cities[1] = "Berlin"  # ❌ Tuples are immutable

# Count how many times a particular city appears
print("Count of 'Paris':", cities.count("Paris"))

# Find the index of a given city
print("Index of 'Tokyo':", cities.index("Tokyo"))

# Iterate and print cities in uppercase
print("Cities in uppercase:")
for city in cities:
    print(city.upper())
```

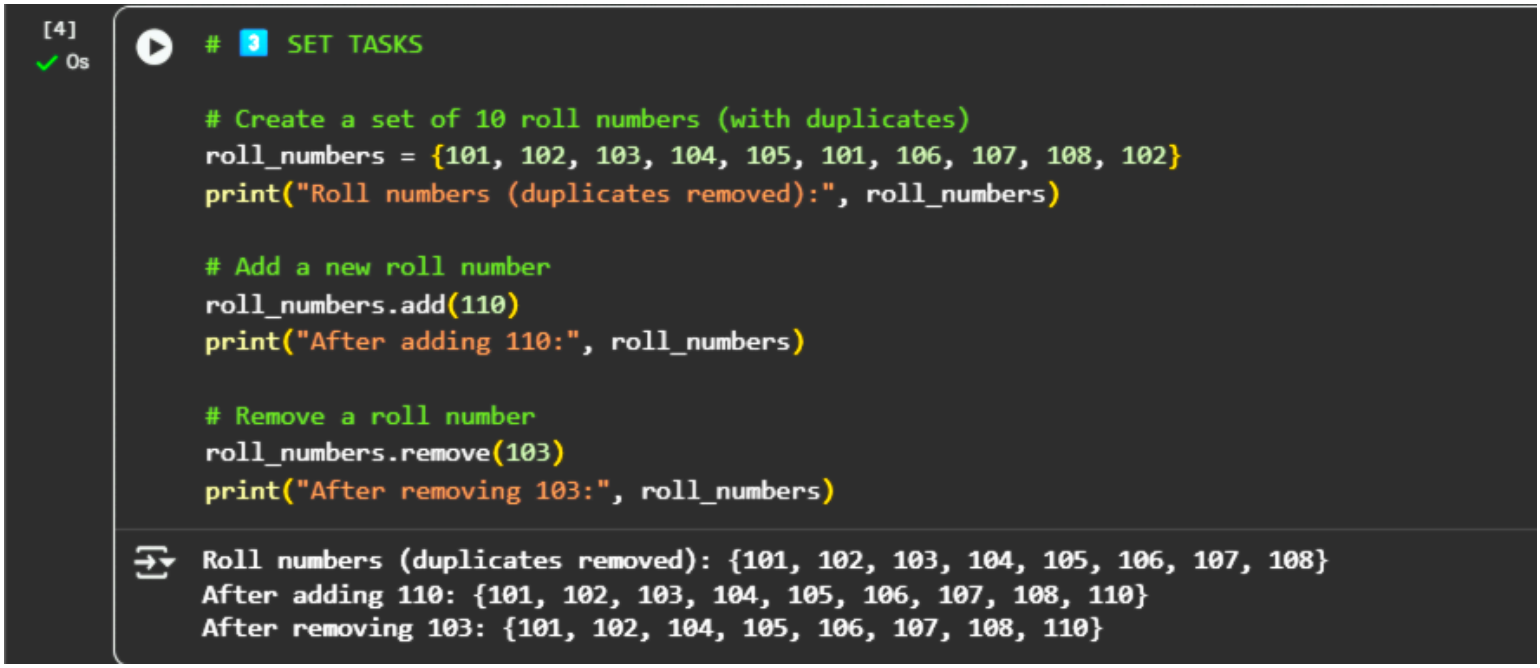
```
First city: New York
Last city: Paris
Count of 'Paris': 2
Index of 'Tokyo': 2
Cities in uppercase:
NEW YORK
PARIS
TOKYO
LONDON
PARIS
```

 # SET TASKS

 # Create a set of 10 roll numbers (with duplicates)
roll_numbers = {101, 102, 103, 104, 105, 101, 106, 107, 108, 102}
print("Roll numbers (duplicates removed):", roll_numbers)

 # Add a new roll number
roll_numbers.add(110)
print("After adding 110:", roll_numbers)

 # Remove a roll number
roll_numbers.remove(103)
print("After removing 103:", roll_numbers)



```
# DICTIONARY TASKS

# Create a dictionary of 5 students
marks = {
    "Alice": 85,
    "Bob": 92,
    "Charlie": 78,
    "David": 90,
    "Eva": 88
}

# Access the marks of a particular student
print("Bob's marks:", marks["Bob"])

# Add a new student and update existing student's marks
marks["Frank"] = 95
marks["Alice"] = 89
print("Updated dictionary:", marks)

# Delete one student
del marks["Charlie"]
print("After deleting Charlie:", marks)

# Iterate and print keys, values, and both
print("Keys:")
for name in marks.keys():
    print(name)
print("Values:")
for score in marks.values():
    print(score)
print("Keys and Values:")
for name, score in marks.items():
    print(name, ":", score)

# Find the student with highest marks
highest_student = max(marks, key=marks.get)
print("Topper:", highest_student, "with marks", marks[highest_student])
```

[5]
✓ 0s

```python
# 4️⃣ DICTIONARY TASKS

# Create a dictionary of 5 students
marks = {
    "Alice": 85,
    "Bob": 92,
    "Charlie": 78,
    "David": 90,
    "Eva": 88
}

# Access the marks of a particular student
print("Bob's marks:", marks["Bob"])

# Add a new student and update existing student's marks
marks["Frank"] = 95
marks["Alice"] = 89
print("Updated dictionary:", marks)

# Delete one student
del marks["Charlie"]
print("After deleting Charlie:", marks)

# Iterate and print keys, values, and both
print("Keys:")
for name in marks.keys():
    print(name)

print("Values:")
for score in marks.values():
    print(score)

print("Keys and Values:")
for name, score in marks.items():
    print(name, ":", score)

# Find the student with highest marks
highest_student = max(marks, key=marks.get)
print("Topper:", highest_student, "with marks", marks[highest_student])
```

```
Bob's marks: 92
Updated dictionary: {'Alice': 89, 'Bob': 92, 'Charlie': 78, 'David': 90, 'Eva': 88, 'Frank': 95}
After deleting Charlie: {'Alice': 89, 'Bob': 92, 'David': 90, 'Eva': 88, 'Frank': 95}
Keys:
Alice
Bob
David
Eva
Frank
Values:
89
92
90
88
95
Keys and Values:
Alice : 89
Bob : 92
David : 90
Eva : 88
Frank : 95
Topper: Frank with marks 95
```