**Lab-2**

**Name:Rhythm Kachhwaha**

**SAP:500107218**

**Batch - 4 AIML**

**1. Create and access tuples.**

**Create a tuple of colors.**

**Access elements using indexing.**

**Try to modify an element in the tuple (to demonstrate immutability).**

**Find the number of occurrences of a specific element in the tuple.**

#Q1

mytuple = ("red","blue","green","orange","violet","blue")

#indexing

mytuple[3] #4th element

#modifying

#mytuple[0] = "blue"

#print(mytuple) #wont be modifiend and show error

print(mytuple.count("blue"))



**Q2. Create and manipulate dictionaries.**

**Create a dictionary to store information about a person (name, age, city).**

**Access values using keys.**

**Add a new key-value pair to the dictionary.**

**Modify an existing value.**

**Check if a key exists in the dictionary.**

**Get a list of all keys and values.**

#Q2

mydict = {'name':"Rhythm",'age':20,'city':"Jodhpur"}

# Accessing the value of the key 'name'

print(mydict['name'])  # Output: Rhythm

# Adding a new key-value pair ('profession', 'student')

mydict['profession'] = "student"

print(mydict)

# Modifying the value of 'age'

mydict['age'] = 21

print(mydict)

# Check if the key 'city' exists

if 'city' in mydict:

print("Key 'city' exists.")

# Getting a list of all keys

keys = mydict.keys()

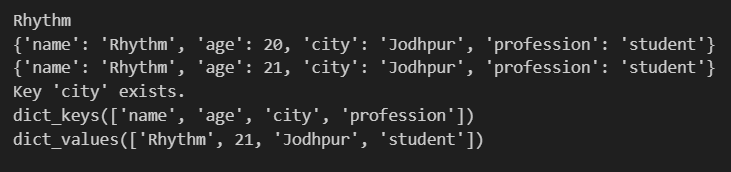
print(keys)  # Output: dict\_keys(['name', 'age', 'city', 'profession'])

# Getting a list of all values

values = mydict.values()

print(values)

**Output:**



**Q3. Demonstrate the difference between mutable and immutable data types.**

**Create a list and a tuple.**

**Try to modify an element in both the list and the tuple.**

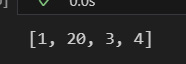
**Observe the results and explain the difference.**

#Q3

my\_list = [1, 2, 3, 4]

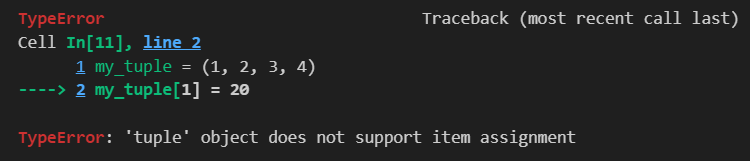
my\_list[1] = 20

print(my\_list)



my\_tuple = (1, 2, 3, 4)

my\_tuple[1] = 20



Lists (Mutable): You can modify, add, or remove elements in a list. After changing an element, the list itself is updated.

Tuples (Immutable): Once a tuple is created, you cannot change its elements. Any attempt to do so will result in an error, indicating that the data cannot be modified after creation. This immutability makes tuples more efficient and safer for read-only purposes.

**Q4. Create a program to print the multiplication table of a number.**

**Take a number as input from the user.**

**Use a for loop to iterate from 1 to 10.**

**Calculate the product of the input number and the current iteration.**

**Print the multiplication table.**

#Q4

# Taking a number as input from the user

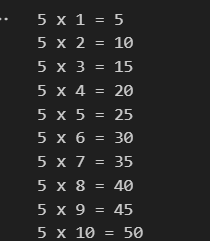
number = int(input("Enter a number: "))

# Using a for loop to iterate from 1 to 10

for i in range(1, 11):

    product = number \* i

    print(f"{number} x {i} = {product}")



**Q5. Create a program to find the factorial of a number using a loop and conditional statements.**

#Q5

number = int(input("Enter a number: "))

factorial = 1

if number < 0:

    print("Factorial is not defined for negative numbers.")

elif number == 0 or number == 1:

    print(f"The factorial of {number} is 1.")

else:

    for i in range(1, number + 1):

        factorial \*= i

    print(f"The factorial of {number} is {factorial}.")



**Q6. Write a program to check if a given number is prime.**

#Q6

number = int(input("Enter a number: "))

# Function to check if the number is prime

def is\_prime(n):

    if n <= 1:

        return False

    elif n <= 3:

        return True

    elif n % 2 == 0 or n % 3 == 0:

        return False

    i = 5

    while i \* i <= n:

        if n % i == 0 or n % (i + 2) == 0:

            return False

        i += 6

    return True

if is\_prime(number):

    print(f"{number} is a prime number.")

else:

    print(f"{number} is not a prime number.")



**Q7. Create a program to find the sum of all even numbers between 1 and 100.**

#Q7

sum = 0

for num in range(1,101):

    if num % 2 == 0:

        sum += num

print(f"The sum of all even numbers between 1 and 100 is {sum}.")



**Q8. Implement a simple calculator using conditional statements and loops.**

#Q8

# Function to perform calculations

def calculate():

    while True:

        # Display menu

        print("\nSimple Calculator")

        print("1. Addition")

        print("2. Subtraction")

        print("3. Multiplication")

        print("4. Division")

        print("5. Exit")

        # Take user choice

        choice = input("Enter choice (1/2/3/4/5): ")

        # Exit the loop if the user chooses to exit

        if choice == '5':

            print("Exiting the calculator. Goodbye!")

            break

        # Validate choice

        if choice in ['1', '2', '3', '4']:

            # Take input numbers

            try:

                num1 = float(input("Enter first number: "))

                num2 = float(input("Enter second number: "))

            except ValueError:

                print("Invalid input. Please enter numeric values.")

                continue

            # Perform calculation based on user choice

            if choice == '1':

                result = num1 + num2

                print(f"{num1} + {num2} = {result}")

            elif choice == '2':

                result = num1 - num2

                print(f"{num1} - {num2} = {result}")

            elif choice == '3':

                result = num1 \* num2

                print(f"{num1} \* {num2} = {result}")

            elif choice == '4':

                if num2 != 0:

                    result = num1 / num2

                    print(f"{num1} / {num2} = {result}")

                else:

                    print("Error! Division by zero is not allowed.")

        else:

            print("Invalid choice. Please select a valid option.")

# Call the calculator function

calculate()

