LAB-2

Name-Ningaraddi

SapId-500110910

Q1. Create and access tuples.

Create a tuple of colors.

```
colors = ('red', 'blue', 'green', 'yellow', 'blue')
```

✓ 0.0s Python

Access elements using indexing.

```
# Accessing the first element
first_color = colors[0]

# Accessing the second element
second_color = colors[1]

# Accessing the last element
last_color = colors[-1]
```

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

```
# Attempting to modify  element
try:

# Count the occurrences of 'blue'
print( colors.count('blue'))
print( colors.count('red'))
print( colors.count('green'))
print( colors.count('yellow'))
```

✓ 0.0s Pytho

```
2
1
1
1
```

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

Q2. Create and manipulate dictionaries.

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

```python
person = {
    'name': 'Ningaraddi',
    'age': 21,
    'city': 'Dehradun'
}
```

✓ 0.0s Python

Access values using keys.

```python
# Accessing values using keys
name = person['name']
age = person['age']
city = person['city']
print(name)
print(age)
print(city)
```

✓ 0.0s Python

```
Alice
30
New York
```

Add a new key-value pair to the dictionary.

```python

person['occupation'] = 'Engineer'
```

Python

Modify an existing value.

```python
# Modifying an existing value
person['age'] = 31

```

Python

Check if a key exists in the dictionary.

```python
# Check if a key exists
has_city_key = 'city' in person  # True
has_country_key = 'country' in person  # False
print(has_city_key)
print(has_country_key)
```

✓ 0.0s Python

```
True
False
```

Get a list of all keys and values.

```python
# Getting a list of all keys
keys = list(person.keys())
print(keys)

# Get a list of all values
values = list(person.values())
print(values)

# Get a list of all key-value pairs
items = list(person.items())
print(items)
```

✓ 0.0s Python

```
['name', 'age', 'city', 'occupation']
['Alice', 30, 'New York', 'Engineer']
[('name', 'Alice'), ('age', 30), ('city', 'New York'), ('occupation', 'Engineer')]
```

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

Create a list and a tuple.

```python
# Creating a list
my_list = [1, 2, 3, 4, 5]

# Creating a tuple
my_tuple = (1, 2, 3, 4, 5)
```

✓ 0.0s Python

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

```python
# Trying to modify an element in the tuple
try:
    my_tuple[2] = 10
except TypeError as e:
    print(f"Error: {e}")
# Modifying an element in the list
my_list[2] = 10
```

✓ 0.0s Python

```
Error: 'tuple' object does not support item assignment
```

Observe the results and explain the difference.

```python
# Creating a list and a tuple
my_list = [1, 2, 3, 4, 5]
my_tuple = (1, 2, 3, 4, 5)

# Modifying an element in the list
my_list[2] = 10

# Trying to modify an element in the tuple
try:
    my_tuple[2] = 10
except TypeError as e:
    print(f"Error: {e}")

print(f"Modified list: {my_list}")
print(f"Original tuple: {my_tuple}")
```

✓ 0.0s Python

```
Error: 'tuple' object does not support item assignment
Modified list: [1, 2, 10, 4, 5]
Original tuple: (1, 2, 3, 4, 5)
```

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.

```python
def multiplication_table(number):
    for i in range(1, 11):
        product = number * i
        # Print the result
        print(f"{number} x {i} = {product}")

# Main part of the program
if __name__ == "__main__":
    # Take a number as input from the user
    try:
        num = int(input("Enter a number to print its multiplication table: "))
        # Print the multiplication table for the input number
        print_multiplication_table(num)
    except ValueError:
        print("Please enter a valid integer.")
```

[21] ✓ 2.9s Python

```
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
```

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

```python
def factorial(number):
    if number < 0:
        return "Factorial is not defined for negative numbers"
    elif number == 0:
        return 1
    else:
        result = 1
        for i in range(1, number + 1):
            result *= i
        return result

# Main part of the program
if __name__ == "__main__":
    try:
        num = int(input("Enter a number to find its factorial: "))

        fact = factorial(num)
        print(f"The factorial of {num} is: {fact}")
    except ValueError:
        print("Please enter a valid integer.")
```

[22] ✓ 1.9s Python

```
The factorial of 5 is: 120
```

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

```python
def is_prime(number):
    if number <= 1:
        return False
    if number == 2:
        return True
    if number % 2 == 0:
        return False


    for i in range(3, int(number**0.5) + 1, 2):
        if number % i == 0:
            return False

    return True


if __name__ == "__main__":
    try:
        num = int(input("Enter a number to check if it is prime: "))
        # Check if the number is prime and print the result
        if is_prime(num):
            print(f"{num} is a prime number.")
        else:
            print(f"{num} is not a prime number.")
    except ValueError:
        print("Please enter a valid integer.")
```

[24] ✓ 1.9s Python

```
77 is not a prime number.
```

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

```python
total_sum = 0

for number in range(1, 101):
    if number % 2 == 0:  # Check if the number is even
        total_sum += number  # Add the even number to the total sum

# Print the result
print(f"The sum of all even numbers between 1 and 100 is: {total_sum}")
```

✓ 0.0s Python

```
The sum of all even numbers between 1 and 100 is: 2550
```

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

```python
def calculator():
    while True:

        print("Simple Calculator")
        print("1. Addition")
        print("2. Subtraction")
        print("3. Multiplication")
        print("4. Division")
        print("5. Exit")

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

        # Check if the user wants to exit
        if choice == '5':
            print("Exiting the calculator. Goodbye!")
            break

        # Check if the choice is valid
        if choice in ['1', '2', '3', '4']:
            try:

                num1 = float(input("Enter the first number: "))
                num2 = float(input("Enter the second number: "))
```

```
                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':
                    # Handle division by zero
                    if num2 == 0:
                        print("Error: Division by zero is not allowed.")
                    else:
                        result = num1 / num2
                        print(f"{num1} / {num2} = {result}")

            except ValueError:
                print("Invalid input. Please enter numeric values.")

        else:
            print("Invalid choice. Please choose a valid operation.")
```

```
# Call the calculator function
calculator()
```

[28] ✓ 13.6s Python

```
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
300.0 * 400.0 = 120000.0
Simple Calculator
1. Addition
2. Subtraction
3. Multiplication
4. Division
5. Exit
Exiting the calculator. Goodbye!
```

+ Code + Markdown