Student Details

Student Name : Monal Ambwani
SAP ID : 590022987
Batch : 17
Course : B.Tech CSE
Subject : Python Programming
Semester : 2

Experiment 3: Loops in Python

Aim

To study and implement looping statements in Python.

Theory

Loops are used to execute a block of code repeatedly until a certain condition is satisfied.

- **for loop**: Used when the number of iterations is known.
- **while loop**: Used when the number of iterations depends on a condition.

Program Codes

1. Factorial of a Number

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

fact = 1
for i in range(1, n + 1):
    fact *= i

print("Factorial =", fact)
```

2. Armstrong Number

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

temp = num
total = 0

while temp > 0:
    digit = temp % 10
    total += digit ** 3
    temp //= 10

if total == num:
    print("Armstrong Number")
else:
    print("Not an Armstrong Number")
```

---

3. Fibonacci Series

```
n = int(input("Enter number of terms: "))

a, b = 0, 1
for i in range(n):
    print(a, end=" ")
    a, b = b, a + b
```

---

4. Prime Number Check

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

if num > 1:
    for i in range(2, num):
        if num % i == 0:
            print("Not a Prime Number")
            break
    else:
        print("Prime Number")
else:
    print("Not a Prime Number")
```

---

5. Palindrome Number

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

rev = 0
temp = num

while temp > 0:
    rev = rev * 10 + temp % 10
    temp //= 10

if rev == num:
    print("Palindrome Number")
else:
    print("Not a Palindrome")
```

---

6. Sum of Digits

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

total = 0
while num > 0:
    total += num % 10
    num //= 10

print("Sum of digits =", total)
```

---

7. Numbers Divisible by 5 or 7

```
count = 0

for i in range(1, 101):
    if i % 5 == 0 or i % 7 == 0:
        print(i, end=" ")
        count += 1

print("\nCount =", count)
```

---

8. Lowercase to Uppercase

```
text = input("Enter a string: ")

result = ""
for ch in text:
    result += ch.upper()

print("Uppercase:", result)
```

---

9. Multiplication Table

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

```
for i in range(1, 11):
    print(num, "*", i, "=", num * i)
```

10. Pattern Printing

```
for i in range(5, 0, -1):
    for j in range(1, i + 1):
        print(j, end="")
    for k in range(6 - i):
        print("*", end="")
    for k in range(6 - i):
        print("*", end="")
    for j in range(i, 0, -1):
        print(j, end="")
    print()
```

11. Sum of Series

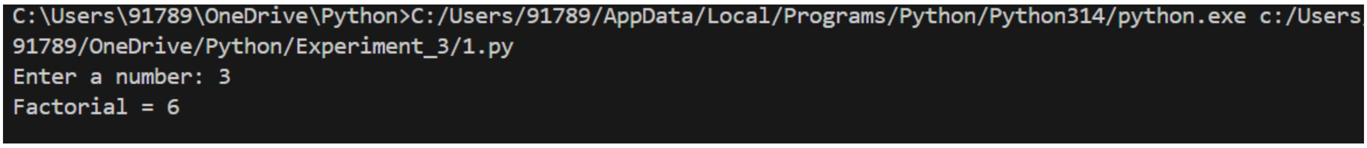
```
n = int(input("Enter value of n: "))

series_sum = 0.0
for i in range(1, n + 1):
    series_sum += 1 / i

print("Sum of series =", series_sum)
```

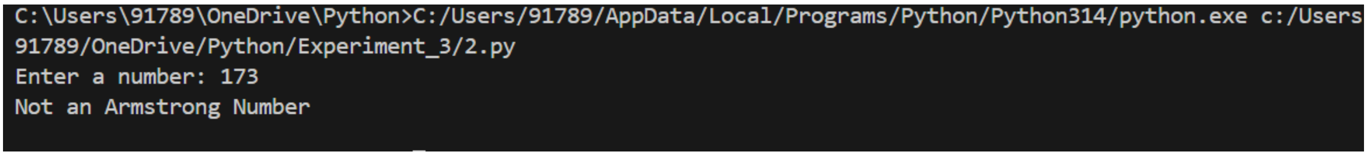
Outputs

1. Factorial of a Number



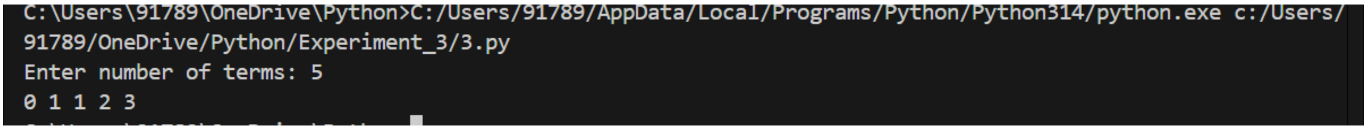
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users
91789/OneDrive/Python/Experiment_3/1.py
Enter a number: 3
Factorial = 6
```

2. Armstrong Number



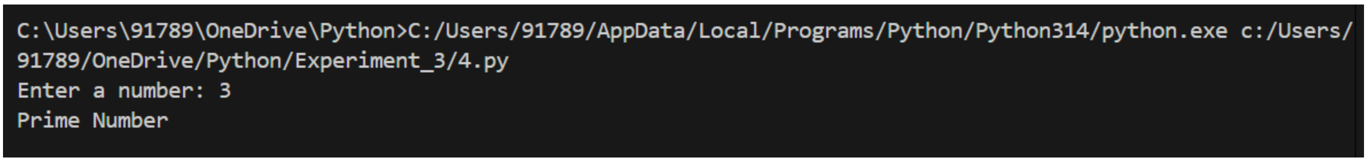
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users
91789/OneDrive/Python/Experiment_3/2.py
Enter a number: 173
Not an Armstrong Number
```

3. Fibonacci Series



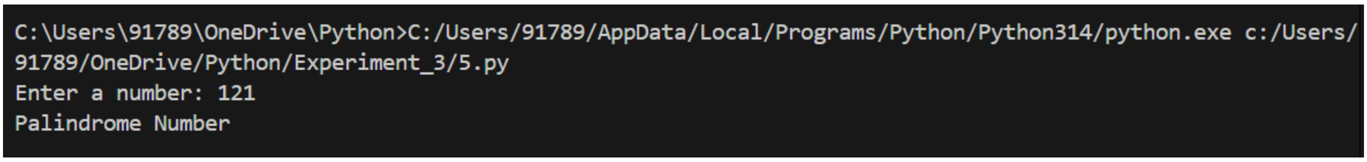
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/3.py
Enter number of terms: 5
0 1 1 2 3
```

4. Prime Number Check



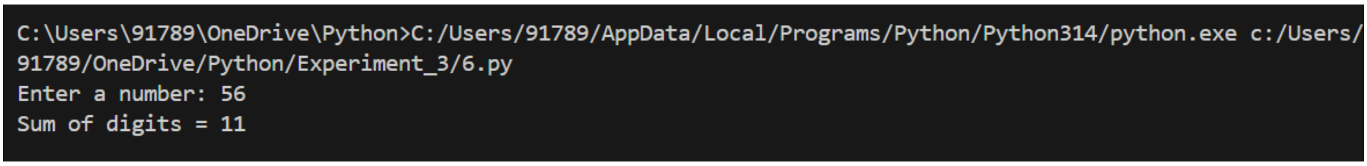
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/4.py
Enter a number: 3
Prime Number
```

5. Palindrome Number



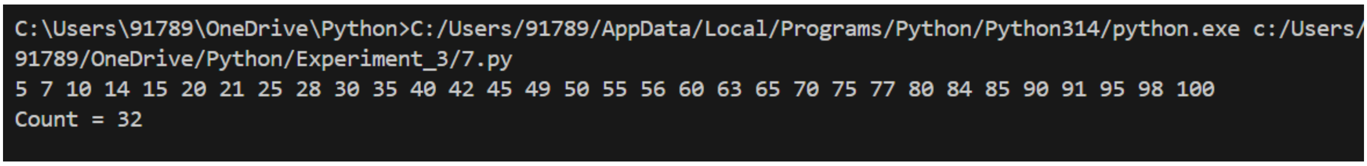
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/5.py
Enter a number: 121
Palindrome Number
```

6. Sum of Digits



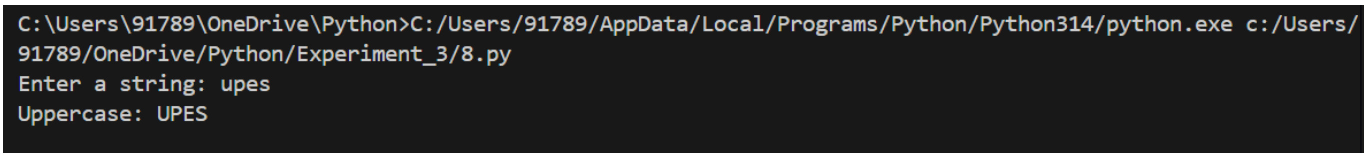
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/6.py
Enter a number: 56
Sum of digits = 11
```

7. Numbers Divisible by 5 or 7



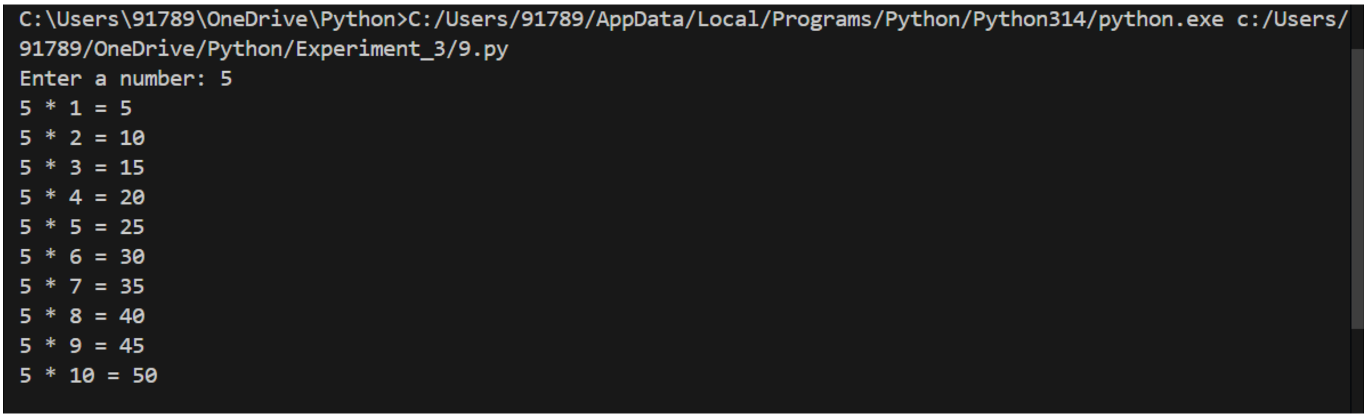
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/7.py
5 7 10 14 15 20 21 25 28 30 35 40 42 45 49 50 55 56 60 63 65 70 75 77 80 84 85 90 91 95 98 100
Count = 32
```

8. Lowercase to Uppercase



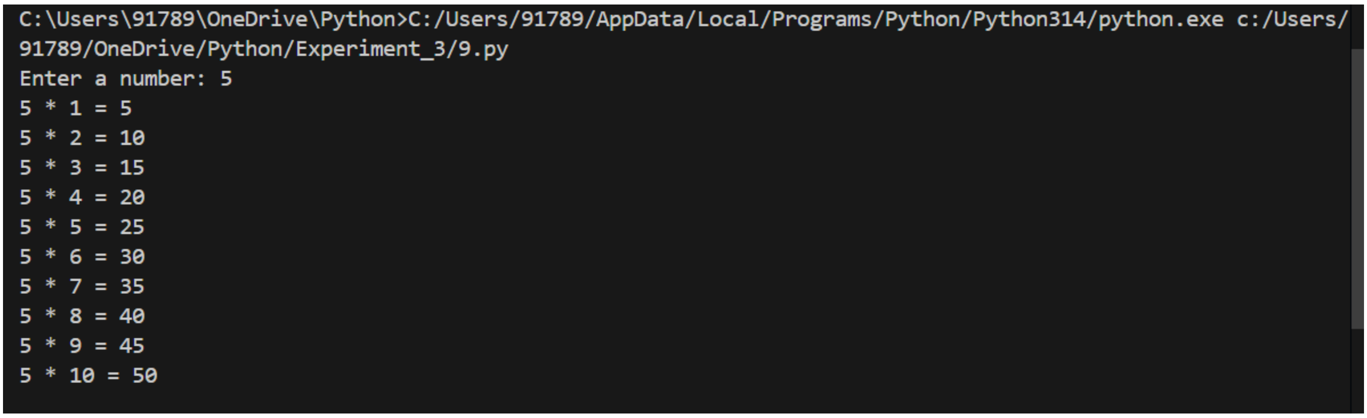
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/8.py
Enter a string: upes
Uppercase: UPES
```

9. Multiplication Table



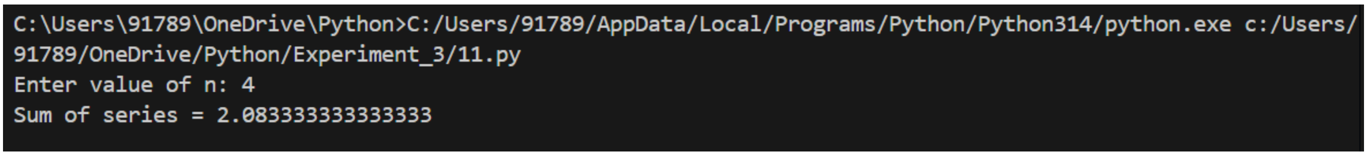
```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/9.py
Enter a number: 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

10. Pattern Printing



```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/9.py
Enter a number: 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
5 * 10 = 50
```

11. Sum of Series



```
C:\Users\91789\OneDrive\Python>C:/Users/91789/AppData/Local/Programs/Python/Python314/python.exe c:/Users/
91789/OneDrive/Python/Experiment_3/11.py
Enter value of n: 4
Sum of series = 2.083333333333333
```

Result

All the programs were executed successfully and looping statements in Python were implemented correctly.