Valluru Sai Balaji

AF0401675

LAB-I:PYTHON OPERATORS

Q.1 Write a python program for arithmetic operators

# Python program for Arithmetic Operators

Num1= 15

Num2= 4

# Addition

Addition = Num1+ Num2

print(Addition)

output

19

# Subtraction

Subtraction = Num1-Num2

print(Subtraction)

output

11

# Multiplication

Multiplication = Num1\*Num2

print(Multiplication)

output

60

# Division

Division = Num1/Num2

print(Division)

output

3.75

# Modulus

Modulus = Num1 % Num2

print(Modulus)

output

3

# Exponentiation

Exponentiation = Num1\*\*Num2

print(Exponentiation)

output

50625

# Floor Division

Floor\_division = Num1// Num2

print(Floor\_division)

output

3

Q.2 Write a python program for assignment operators

# Python program for Assignment Operators

Num1 = 10

Num2 = 3

# Add and assign

Num1 += Num2 # Equivalent to Num1 = Num1 + Num2

print("Num1 += Num2 is", Num1)

output

Num1 += Num2 is 13

# Subtract and assign

Num1 -= Num2 # Equivalent to Num1 = Num1 - Num2

print("Num1 -= Num2 is", Num1)

output

Num1 -= Num2 is 10

# Multiply and assign

Num1 \*= Num2 # Equivalent to Num1 = Num1 \* Num2

print("Num1 \*= Num2 is", Num1)

output

Num1 \*= Num2 is 30

# Divide and assign

Num1 /= Num2 # Equivalent to Num1 = Num1 / Num2

print("Num1 /= Num2 is", Num1)

output

Num1 /= Num2 is 10.0

# Modulus and assign

Num1 %= Num2 # Equivalent to Num1 = Num1 % Num2

print("Num1 %= Num2 is", Num1)

output

Num1 %= Num2 is 1.0

# Exponentiation and assign

Num1 \*\*= Num2 # Equivalent to Num1 = Num1 \*\* Num2

print("Num1 \*\*= Num2 is", Num1)

output

Num1 \*\*= Num2 is 1.0

# Floor division and assign

Num1 //= Num2 # Equivalent to Num1 = Num1 // Num2

print("Num1 //= Num2 is", Num1)

output

Num1 //= Num2 is 0.0

Q.3Write a python  program for Bitwise operators

# Python program for Bitwise Operators

Num1 = 10 # Num1 in binary: 1010

Num2 = 4 # Num2 in binary: 0100

# Bitwise AND

# Performs a bitwise AND operation on Num1 and Num2.

# It compares each bit of Num1 and Num2 and returns a new value where each bit is set to 1 only if both corresponding bits are 1.

bitwise\_and = Num1 & Num2

print(f"Bitwise AND: {Num1} & {Num2} = {bitwise\_and}")

# Bitwise OR

# Performs a bitwise OR operation on Num1 and Num2.

# It compares each bit of Num1 and Num2 and returns a new value where each bit is set to 1 if at least one of the corresponding bits is 1.

bitwise\_or = Num1 | Num2

print(f"Bitwise OR: {Num1} | {Num2} = {bitwise\_or}")

# Bitwise XOR

# Performs a bitwise XOR (exclusive OR) operation on Num1 and Num2.

# It compares each bit of Num1 and Num2 and returns a new value where each bit is set to 1 only if the corresponding bits are different.

bitwise\_xor = Num1 ^ Num2

print(f"Bitwise XOR: {Num1} ^ {Num2} = {bitwise\_xor}")

# Bitwise NOT

# Performs a bitwise NOT operation on Num1.

# It inverts all the bits of Num1, changing 1s to 0s and 0s to 1s.

bitwise\_not = ~Num1

print(f"Bitwise NOT: ~{Num1} = {bitwise\_not}")

# Bitwise right shift

# Performs a right shift operation on Num1.

# It shifts the bits of Num1 to the right by 1 position, effectively dividing the number by 2.

right\_shift = Num1 >> 1

print(f"Right Shift: {Num1} >> 1 = {right\_shift}")

# Bitwise left shift

# Performs a left shift operation on Num1.

# It shifts the bits of Num1 to the left by 1 position, effectively multiplying the number by 2.

left\_shift = Num1 << 1

print(f"Left Shift: {Num1} << 1 = {left\_shift}")

output

Bitwise AND: 10 & 4 = 0

Bitwise OR: 10 | 4 = 14

Bitwise XOR: 10 ^ 4 = 14

Bitwise NOT: ~10 = -11

Right Shift: 10 >> 1 = 5

Left Shift: 10 << 1 = 20

Q.4 Write a python program to calculate greatest of three numbers

# python program to calculate greatest of three numbers

# python program to calculate greatest of three numbers

Num1 = 10

Num2 = 15

Num3 = 20

greatest\_number = Num1<Num2<Num3

print(greatest\_number)

output

True