Q.1. Create two int type variables, apply addition, subtraction, division and multiplications and store the results in variables. Then print the data in the following format by calling the

variables:

First variable is \_\_ & second variable is \_\_.

Addition: \_\_ + \_\_ = \_\_

Subtraction: \_\_ - \_\_ = \_\_

Multiplication: \_\_ \* \_\_ = \_\_

Division: \_\_ / \_\_ = \_\_

###############################################Python Code###################################################

def addition(a,b):

return a+b

def subtraction(a,b):

return a-b

def division(a,b):

return a/b

def multiplication(a,b):

return a\*b

a = 10

b = 11

print(f"Addition: {a} + {b} = {addition(a,b)}")

print(f"Subtraction: {a} - {b} = {subtraction(a,b)}")

print(f"Multiplication: {a} \* {b} = {multiplication(a,b)}")

print(f"Division: {a} / {b} = {division(a,b)}")

################################################Python Code###################################################

Q.2. What is the difference between the following operators:

(i) ‘/’ & ‘//’

(ii) ‘\*\*’ & ‘^’

A. '/' is called slash and used for division whereas '//' is called as floor division. using slash we divide the integer and get the result in either float or integer format. But using '//' if we divide the integer we get the result in integer rounded to the nearest integer value

'\*\*' is used to find the exponent of an integer. '^' means bitwise or in C/python. Hence python chose '\*\*' for exponent.

Q.3. List the logical operators.

A. And,Or, Not are the logical operators

Q.4. Explain right shift operator and left shift operator with examples.

A. The right shift operator (>>) in Python shifts the bits of a binary number to the right, effectively dividing it by 2 raised to the power of the shift count. It fills the vacant positions on the left with the sign bit. For positive numbers, it's 0, and for negative numbers, it's 1.

x = 12 # Binary: 1100

y = x >> 2 # Shift right by 2 positions

# Binary after shifting: 0011

# Decimal: 3

print(y) # Output: 3

The left shift operator (<<) in Python shifts the bits of a binary number to the left, effectively multiplying it by 2 raised to the power of the shift count. It fills the vacant positions on the right with zeros.

x = 3 # Binary: 11

y = x << 2 # Shift left by 2 positions

# Binary after shifting: 1100

# Decimal: 12

print(y) # Output: 12

Q.5. Create a list containing int type data of length 15. Then write a code to check if 10 is

present in the list or not

A.

numbers = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15]

def checkIfPresnt(n:int,arr:list) -> bool:

return n in arr

print(checkIfPresen(10,numbers))