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

# Create two int type variables

num1 = 10

num2 = 5

# Perform arithmetic operations and store results in variables

addition = num1 + num2

subtraction = num1 - num2

multiplication = num1 \* num2

division = num1 / num2

# Print the data in the specified format

print(f"First variable is {num1} & second variable is {num2}.")

print(f"Addition: {num1} + {num2} = {addition}")

print(f"Subtraction: {num1} - {num2} = {subtraction}")

print(f"Multiplication: {num1} \* {num2} = {multiplication}")

print(f"Division: {num1} / {num2} = {division}")

Output:

First variable is 10 & second variable is 5.

Addition: 10 + 5 = 15

Subtraction: 10 - 5 = 5

Multiplication: 10 \* 5 = 50

Division: 10 / 5 = 2.0

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

(i) ‘/’ & ‘//’

(ii) ‘\*\*’ & ‘^’

(i) The '/' operator performs regular division, which results in a float value.

The '//' operator performs integer division, which returns the quotient as an integer, discarding any decimal part.

(ii) The \*\* operator is used for exponentiation (raising a number to a power). For example, 2 \*\* 3 means 2 raised to the power of 3, resulting in 8.

The '^' operator is not a valid operator for exponentiation in Python. Instead, the \*\* operator is used.

Q.3. List the logical operators.

and: Returns True if both operands are True, otherwise False.

or: Returns True if at least one of the operands is True, otherwise False.

not: Returns the opposite of the operand's value (True becomes False, and vice versa).

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

The right shift operator (>>) and left shift operator (<<) are bitwise shift operators.

Right Shift Operator (>>): Shifts the bits of a number to the right by a specified number of positions. The rightmost bits are dropped, and the leftmost positions are filled with zeros.

x = 12 # Binary: 1100

result = x >> 2

print(result) # Output: 3 (Binary: 0011)

Left Shift Operator (<<): Shifts the bits of a number to the left by a specified number of positions. Zeros are added to the right.

Example:

x = 5 # Binary: 0101

result = x << 2

print(result) # Output: 20 (Binary: 10100)

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

# Create a list containing int type data of length 15

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

# Check if 10 is present in the list

if 10 in my\_list:

print("10 is present in the list.")

else:

print("10 is not present in the list.")

10 is present in the list.