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:

In [1]:

```
# The two int type variables is.
num1 = 10
num2 = 5

# Perform addition, subtraction, division, and multiplication
addition_result = num1 + num2
subtraction_result = num1 - num2
division_result = num1 / num2
multiplication_result = num1 * num2

# Print the data in the specified format
print(f"Number 1: {num1}")
print(f"Number 2: {num2}")
print(f"Addition Result: {addition_result}")
print(f"Subtraction Result: {subtraction_result}")
print(f"Division Result: {division_result}")
print(f"Multiplication Result: {multiplication_result}")
```

```
Number 1: 10
Number 2: 5
Addition Result: 15
Subtraction Result: 5
Division Result: 2.0
Multiplication Result: 50
```

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

(i) '/' & '//'

(ii) '**' & '^'

Ans:- (1)'/' & '//':- (a)'/': The forward slash '/' is the standard division operator in most programming languages, including Python. When used with two numeric operands, it performs floating-point division, which means it returns a floating-point result with decimal places. For example, 5 / 2 would result in 2.5 (b)'//': The double forward slash '//' is the floor division operator in Python. It performs division but rounds the result down to the nearest whole number (floor value). It discards the decimal part of the result and returns an integer. For example, 5 // 2 would result in 2, as it discards the decimal part.

```
    (2)‘**’ & ‘^’:-
    (a)'**'': The double asterisk '' is the exponentiation operator in Python. It i
    s used to raise a number to a certain power. For example, 2 ** 3 would result i
    n 8 because it calculates 2 raised to the power of 3.
    (b)'^' : The caret symbol '^' is not an exponentiation operator in Python. Inst
```

3. List the logical operators.

Ans:- Logical operators are used in programming and mathematics to combine and manipulate logical values (true or false). The most common logical operators are: (1) AND (&&): It returns true if both operands are true; otherwise, it returns false. (2) OR (||): It returns true if at least one of the operands is true; if both operands are false, it returns false. (3) NOT (!): It reverses the logical state of its operand. If the operand is true, it returns false, and if the operand is false, it returns true.

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

```
Ans:
    (1) Right Shift Operator (>>):
The right shift operator shifts the binary representation of a number to the right by
a specified number of positions, effectively dividing the number by 2 raised to the
power of the shift count. This is equivalent to performing an integer division by 2
for each right shift.
For Example:-
    Let's consider an 8-bit binary number 10101110, which is equivalent to the decimal
number 174.

Binary representation: 10101110
Decimal representation: 174

If we perform a right shift by 2 positions (num >> 2), the binary representation will
be shifted to the right as follows:

Original binary: 10101110
After right shift: 00101011 (decimal equivalent: 43)

So, 174 >> 2 is equal to 43.

    (2)Left Shift Operator (<<):
The left shift operator shifts the binary representation of a number to the left by a
specified number of positions, effectively multiplying the number by 2 raised to the
power of the shift count. This is equivalent to performing an integer multiplication
by 2 for each left shift.
For Example:-
    Let's consider an 8-bit binary number 00110010, which is equivalent to the decimal
number 50.

Binary representation: 00110010
Decimal representation: 50

If we perform a left shift by 3 positions (num << 3), the binary representation will
be shifted to the left as follows:

Original binary: 00110010
After left shift: 10010000 (decimal equivalent: 160)
```

```
So, 50 << 3 is equal to 160.
```

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.

In [2]:

```
# Creating a list of length 15 with integer type data
my_list = [2, 5, 8, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65]

if 10 in my_list:
    print("Number 10 is present in the list.")
else:
    print("Number 10 is not present in the list.")
```

```
Number 10 is present in the list.
```

In [ ]:

```

```