**Question 2**:

* / is the division operator in Python, which performs normal division and returns a floating-point number (float) as the result.

// is the floor division operator in Python, which performs division and returns the integer part of the quotient, discarding any fractional part. It always returns an integer, even if the result is technically a floating-point number.

Example: 10/3=3.3333333333; 10//3=3

* \*\* is the exponentiation operator in Python, which raises the left operand to the power of the right operand.

^ is the bitwise XOR operator in Python, which performs bitwise XOR operation between the bits of two operands.

Example: 2\*\*3=8; 2^3=1

**Question 3**:

* and
* or
* not

**Question 4**:

* Right shift operator shifts all the bits of a binary number to the right by a specified number of positions.

num = 8 # Binary: 00001000  
shifted = num >> 2 # Shift right by 2 positions  
print(bin(shifted)) # Output: 0b10 (binary representation of 2)

* left shift operator shifts all the bits of a binary number to the left by a specified number of positions.

num = 2 # Binary: 00000010  
shifted = num << 3 # Shift left by 3 positions  
print(bin(shifted)) # Output: 0b10000 (binary representation of 16)