Syntax of nested statement

```
if (condition 1) :
    if (condition 2) :
        print
        statement of condition 2
    else :
        statement of inner if else ( condition 2)
else :
    statement of outer if else ( condition 1).
```

Eg 3,

```
n = float (input ("enter the number :"))
if (n>=0)
    if (n > 0)
        print ("positive")
    else :
        print ("zero")
else:
    print ("negative")
```

7. Python bitwise operation

- are used to compare (binary) numbers

1. and (&)

- Result is 1 , only if both bits are 1

Eg :- 5 & 3 = 1

2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1

```
  0 1 0 1
  0 0 1 1
& 0 0 0 1   = 1
```

2. OR (|)

- Result is 1, if atleast 1 bit is 1.

Eg :- 7 | 14 = 15

```
  0 0 1 1 1
  0 1 1 1 0
-----------
  0 1 1 1 1  = 15
```

3. XOR (^)

- Result is 1, if bits are different.

Eg :- 5 ^ 3 = 6

```
  0 1 0 1
  0 0 1 1
-----------
  0 1 1 0  = 6
```

4. Signed left shift (<<)

- Shift bits to the left, & fills with '0' on the right.

Eg :- 5 << 1 = 10

2^5 2^4 2^3 2^2 2^1 2^0

32 16 8 4 2 1

-> 5 = 0 0 1 0 1

0 1 0 1 0

8 + 2 = 10

Eg :- 5 << 2 = 20

```
  0 0 1 0 1
  0 0 1 0 1 0
  0 0 1 0 1 0 0
```

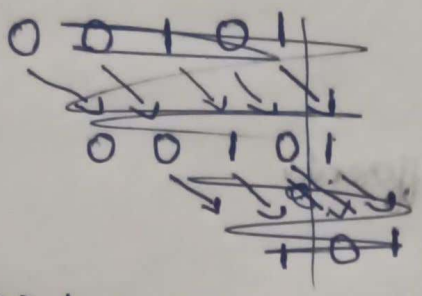
16 + 4 = 20

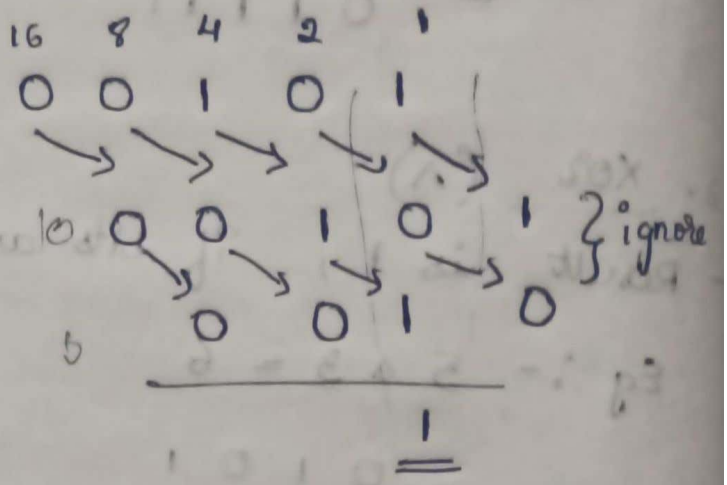
5. Signed right shift (>>)

- Shift bits to the right.
- for +ve no. → fills left side with '0'.
- for -ve no. → fills left side with '1'.

Eg : (positive) →

5 >> 2 = 1





-10 >> 1 = -5