Number system converstion (bit-binary digit)

binary : base(0-1)--> please divide 15/2 & count in reverse order octal:base (0-7)
hexadecimal : base(0-9 & then a-f) when you check ip address you will find these format-->cmd-ipconfig

In [25]:
```
25
```

Out[25]: 25

In [29]:
```
bin(25)
```

Out[29]: '0b11001'

In [33]:
```
0b11001
```

Out[33]: 25

In [35]:
```
int(0b11001)
```

Out[35]: 25

In [37]:
```
bin(35)
```

Out[37]: '0b100011'

In [39]:
```
int(0b100011)
```

Out[39]: 35

In [41]:
```
bin(20)
```

Out[41]: '0b10100'

In [43]:
```
int(0b10100)
```

Out[43]: 20

In [45]:
```
oct(15)
```

Out[45]: '0o17'

In [47]:
```
0o17
```

Out[47]: 15

In [49]:
```
hex(9)
```

Out[49]: '0x9'

In [51]:
```
0x9
```

Out[51]: 9

In [53]:
```
0xf
```

Out[53]: 15

In [55]:
```
hex(10)
```

Out[55]: '0xa'

In [57]:
```
hex(25)
```

Out[57]: '0x19'

In [59]:
```
0x15
```

Out[59]: 21

In [61]:
```
0x19
```

Out[61]: 25

Swap variable in python

In [64]:
```
a = 5
b = 6
```

In [66]:
```
a = b
b = a
```

In [68]:
```
a,b = b,a
```

In [70]:
```
print(a)
print(b)
```

```
6
6
```

In [72]:
```
a = 5
b = 6
```

In [74]:
```
a,b = b,a
```

In [76]:
```
print(a)
print(b)
```

```
6
5
```

In [78]:
```
a1 = 7
b1 = 8
```

In [80]:
```
temp = a1
a1 = b1
b1 = temp
```

In [82]:
```
print(a1)
print(b1)
```

```
8
7
```

In [84]:
```
a2 = 4
b2 = 5
```

In [86]:
```
a2 = a2+b2
b2 = a2-b2
a2 = a2-b2
```

In [88]:
```
print(a2)
print(b2)
```

```
5
4
```

In [90]:
```
print(0b101)
print(0b110)
```

```
5
6
```

In [92]:
```
print(bin(11))
print(0b1011)
```

```
0b1011
11
```

In [94]:
```
a2 = a2 ^ b2
b2 = a2 ^ b2
a2 = a2 ^ b2
```

In [96]:
```
print(a2)
print(b2)
```

```
4
5
```

In [98]:
```
print(a2)
print(b2)
```

```
4
5
```

In [100…
```
a2,b2 = b2,a2
```

In [102…
```
print(a2)
print(b2)
```

```
5
4
```

BITWISE OPERATOR

- We have 6 operators complement(~) || And(&) || OR(|) || XOR(^) || Left Shift(<<) || Right Shift(>>)

In [110…
```
print(bin(12))
print(bin(13))
```

```
0b1100
0b1101
```

complement --> you will get this key below esc character

12 ==> 1100 || first thing we need to understand what is mean by complement. complement means it will do reverse of the binary format i.e. - ~0 it will give you 1 ~1 it will give 0 12 binary format is 00001100 (complement of ~00001100 reverse the number - 11110011 which is (-13)

but the question is why we got -13 to understand this concept (we have concept of 2's complement 2's complement mean (1's complement + 1) in the system we can store +Ve number but how to store -ve number

lets understand binary form of 13 - 00001101 + 1

In [114…
```
~45
```

Out[114… -46

In [116…
```
~6
```

Out[116… -7

In [118…
```
~-6
```

Out[118… 5

In [120…
```
~-2
```

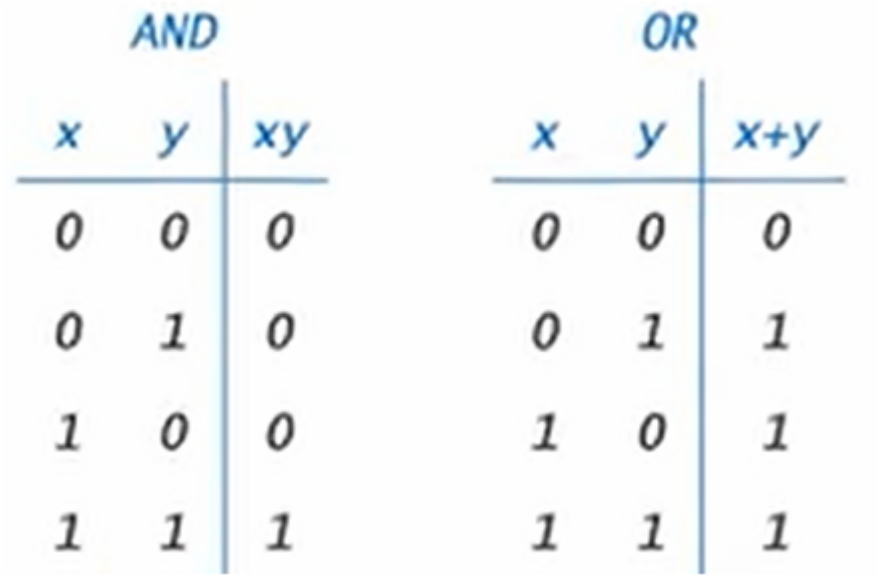
Out[120… 1

In [122…
```
~-1
```

Out[122… 0

Bit wise and operator

AND - LOGICAL OPERATOR ||| & - BITWISE AND OPERATOR (we know that 1 & 1 is 1) 12 - 00001100 13 - 00001101 when we are add both then then outut we will get as 12

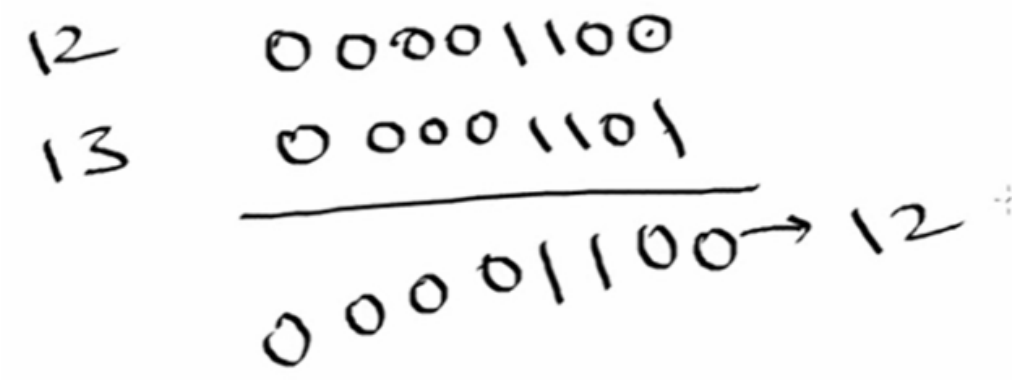


AND

x	y	xy
0	0	0
0	1	0
1	0	0
1	1	1

OR

x	y	x+y
0	0	0
0	1	1
1	0	1
1	1	1



In [129…
```
12 & 13
```
Out[129… 12

In [131…
```
1 & 1
```
Out[131… 1

In [133…
```
1 | 0
```
Out[133… 1

In [135…
```
1 & 0
```
Out[135… 0

In [137…
```
12 | 13
```
Out[137… 13

In [143…
```
bin(35)
```
Out[143… '0b100011'

In [145…
```
bin(40)
```

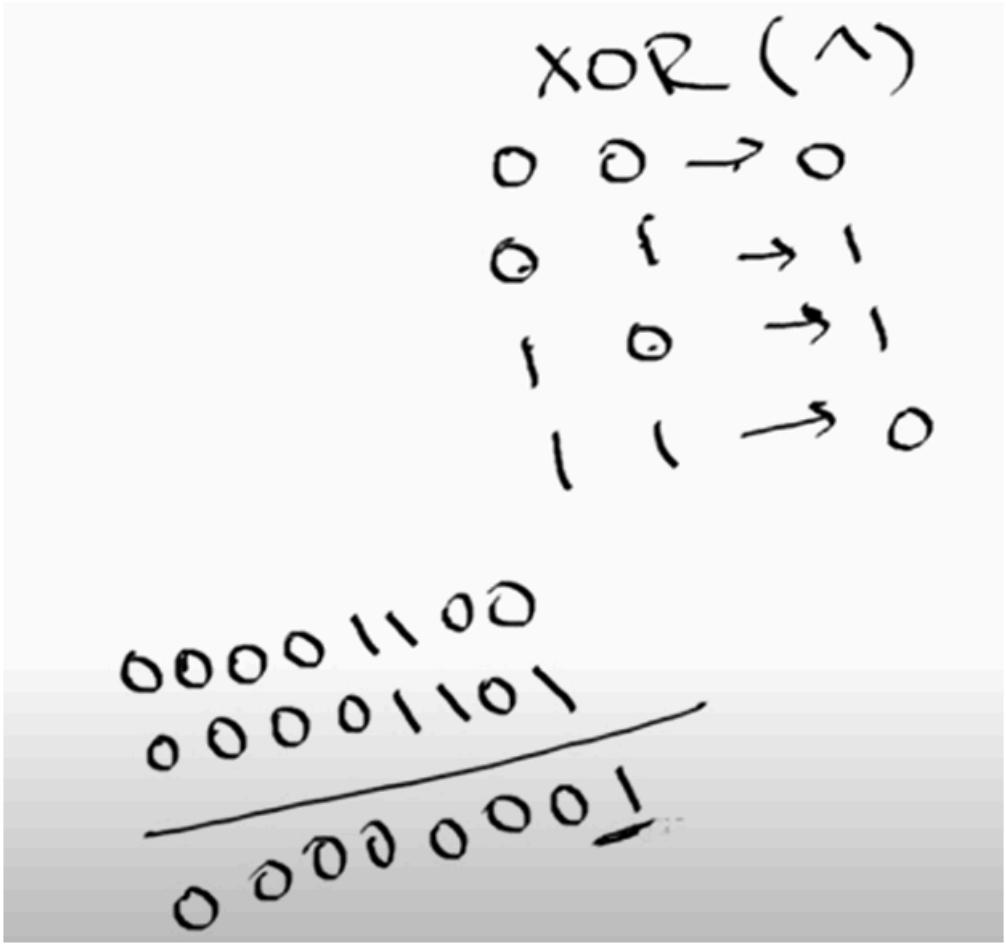
Out[145… '0b101000'

In [139…
```
35 & 40
```

Out[139… 32

In [141…
```
35 | 40
```

Out[141… 43



In [148…
```
12 ^ 13
```

Out[148… 1

In [158…
```
bin(25)
```

Out[158… '0b11001'

In [160…
```
bin(30)
```

Out[160… '0b11110'

In [150…
```
25 ^ 30
```

Out[150… 7

In [152…
```
10 ^ 15
```

Out[152… 5

In [162…
```
int(0b000111)
```

Out[162… 7

In [164…
```
bin(7)
```

Out[164… '0b111'

bitwise leftshift operator

In [ ]:
```
#bit wise left operator bydefault you will take 2 zeros ( )
 #10 binary operator is 1010 | also i can say 1010
```

In [167…
```
10<<2
```

Out[167… 40

In [169…
```
20<<2
```

Out[169… 80

In [171…
```
20<<4
```

Out[171… 320

Bitwise Rightshift operator

In [177…
```
10>>2
```

Out[177… 2

In [181…
```
bin(20)
```

Out[181… '0b10100'

In [179…
```
20>>4
```

Out[179… 1

In [ ]: