Left shift ( << )

A = 8 << 3

Assume 8 bit data,

8 in binary = 00001000 (🡨---------)

Left shift first time

00010000 (msb is removed, and 0 is appended at right side ie. After Lsb

Left shift second time

00100000

Left shift third time

01000000

8 << 3 = (01000000)2 = 64

Bitwise right shift ( >> ) -------🡪

32 >> 3

32 in binary, 8 bit representation

00100000

Right shift first time

00010000 (Lsb is removed, and 0 is appended at left side ie. before msb

Right shift second time

00001000

Right shift third time

00000100

32 >> 3 = (00000100)2 =(4)10

(24 + 22 + 20) = 16 + 4 + 1 = 21 in decimal

20 = 1, 21 = 2, 22 = 4, 23 = 8, 24 = 16, 25 = 32

39 = 32 + 4 + 2 + 1 = (100111)2

(39-32=7,

(25)10 convert to binary: 16 + 8 + 1

(25-16=9, 9-8=1,1-1=0)

1x24 + 1x23 + 0x22 + 0x21 + 1x20

(11001)2

(0 1 2 3……9)

(0 1)

16 0 1 2 …9 A B C D E F

Q R

25 / 2 = 12 1

12 / 2 = 6 0

6 / 2 = 3 0

3 / 2 = 1 1

1 / 2 = 0 1

(25) = 11001

(9) =(1001)=8+1=9