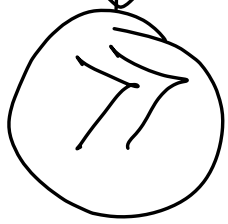
Shifting Operators

(Bitwise Operators)

- left shift
- right shift



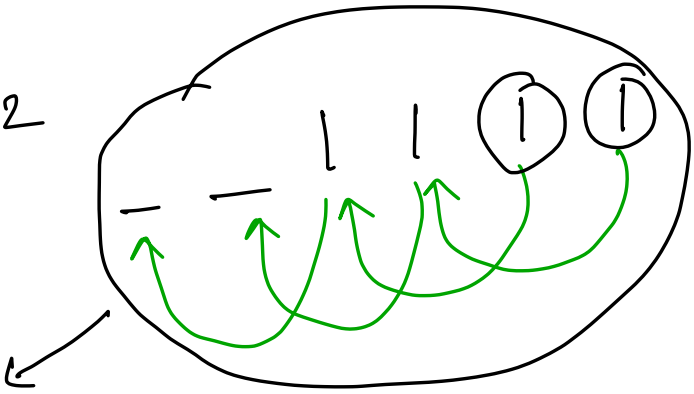
1. Boldness
2. Confident
3. Leadership
4. Responsible

a = 15

a << 2 ⟹

1. Convert into Binary.
2. Shift all the Binary digits left side depending on no. of shifts.

$a = (15)_{10} = (1111)_2$



$(1111\ 00)_2 \rightarrow (60)_{10}$ ← a << 2

15 —2 bit left shift→ 60

Left shift Result ⇒ $(a \times 2^x)$

15×2^2

$= 15 \times 4 = 60$

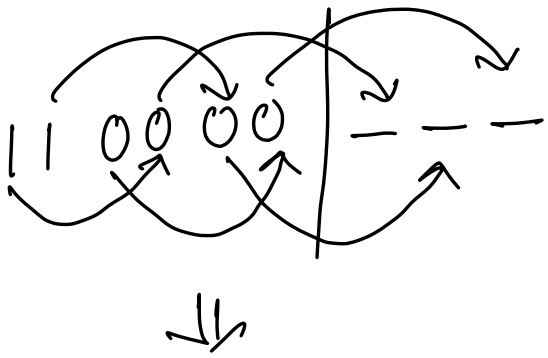
Right shift

a = 48

↓

$(110000)_2$

a >> 3



$6 \Leftarrow (000110)_2 \,|\, 000$

Right shift ⇒ $\frac{a}{2^n}$ formula

$48 \xrightarrow[RS]{3\ bit} 6$

$\frac{48}{2^3} = \frac{48}{8} = 6$

Conditional operator

Syntax

(Expression or variable or constant) ? (statement 1) : (statement 2);

True

False

```
int a=10, b=15;
a>b ? printf("a in greater than b"):
      printf("b in greater than a");
```

Sizeof operator

sizeof(<name of the variable>)

```
int a;
printf("%d in the size of variable a", sizeof(a));
```

4

printf()

→ <stdio.h>

Standard Input Output

<stdlib.h> → Standard Library

* To print Anything. (Function).

* Return

The total Number of Elements / characters it is printing.

---

int → %i / %d

float → %f

character → %c

double → %lf

long int → %ld

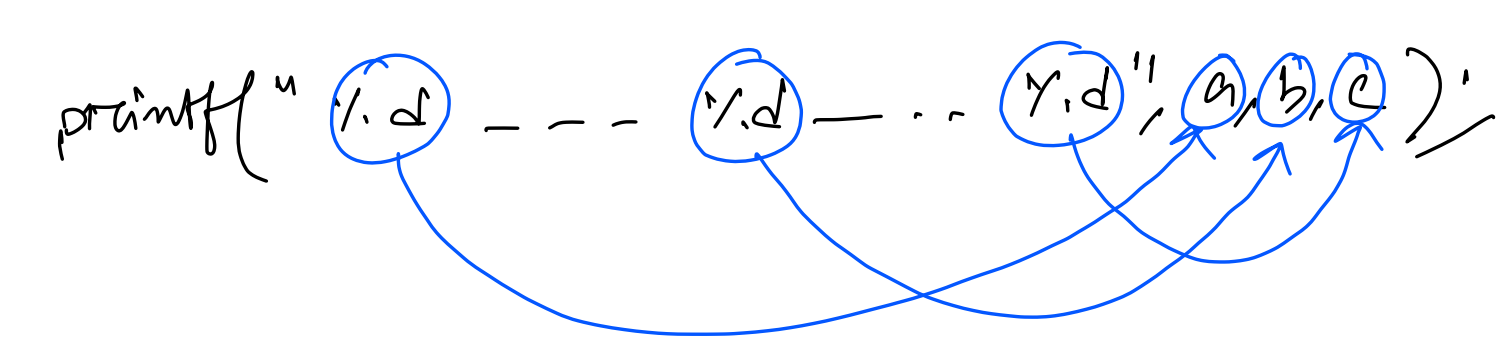
long long int → %lld

Hexadecimal → %x / %X

Octal → %o

0x ——

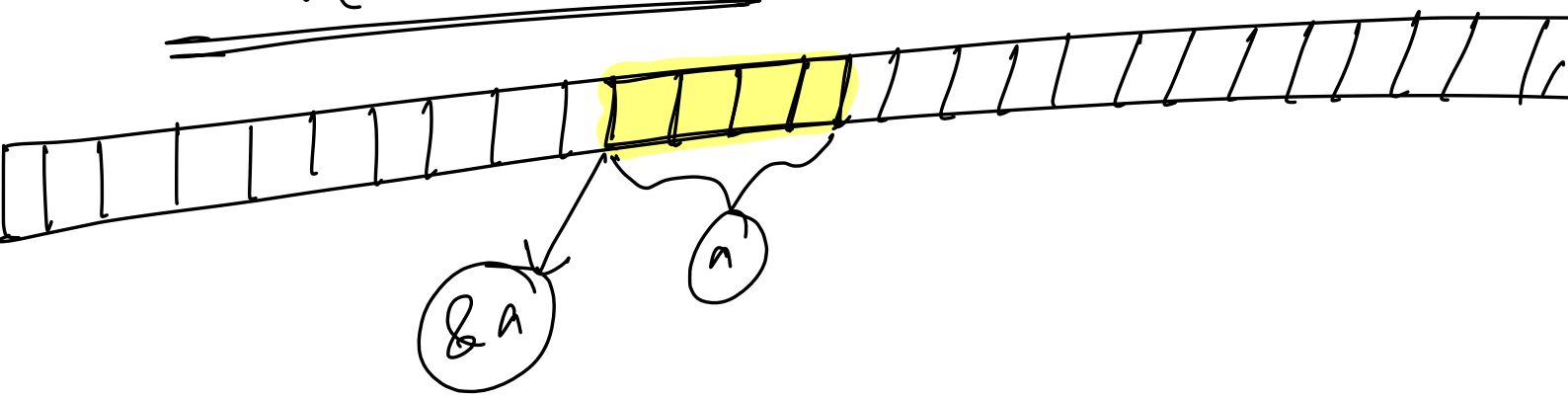
0x ——



scanf()

int a;
scanf(" %d ", &a)

address of operator

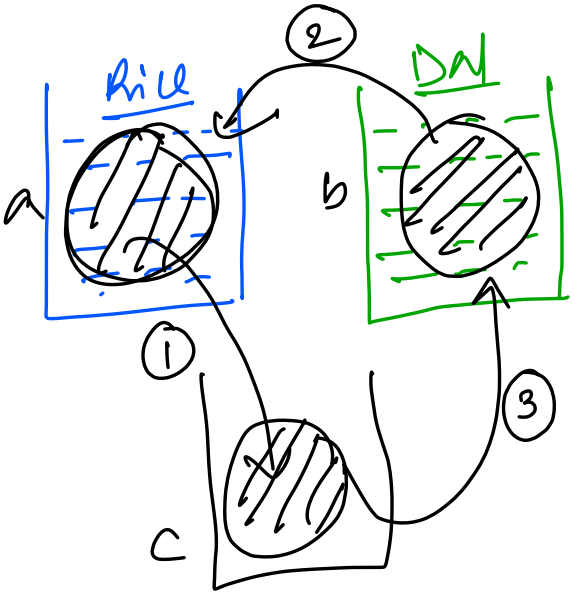


swapping variables.

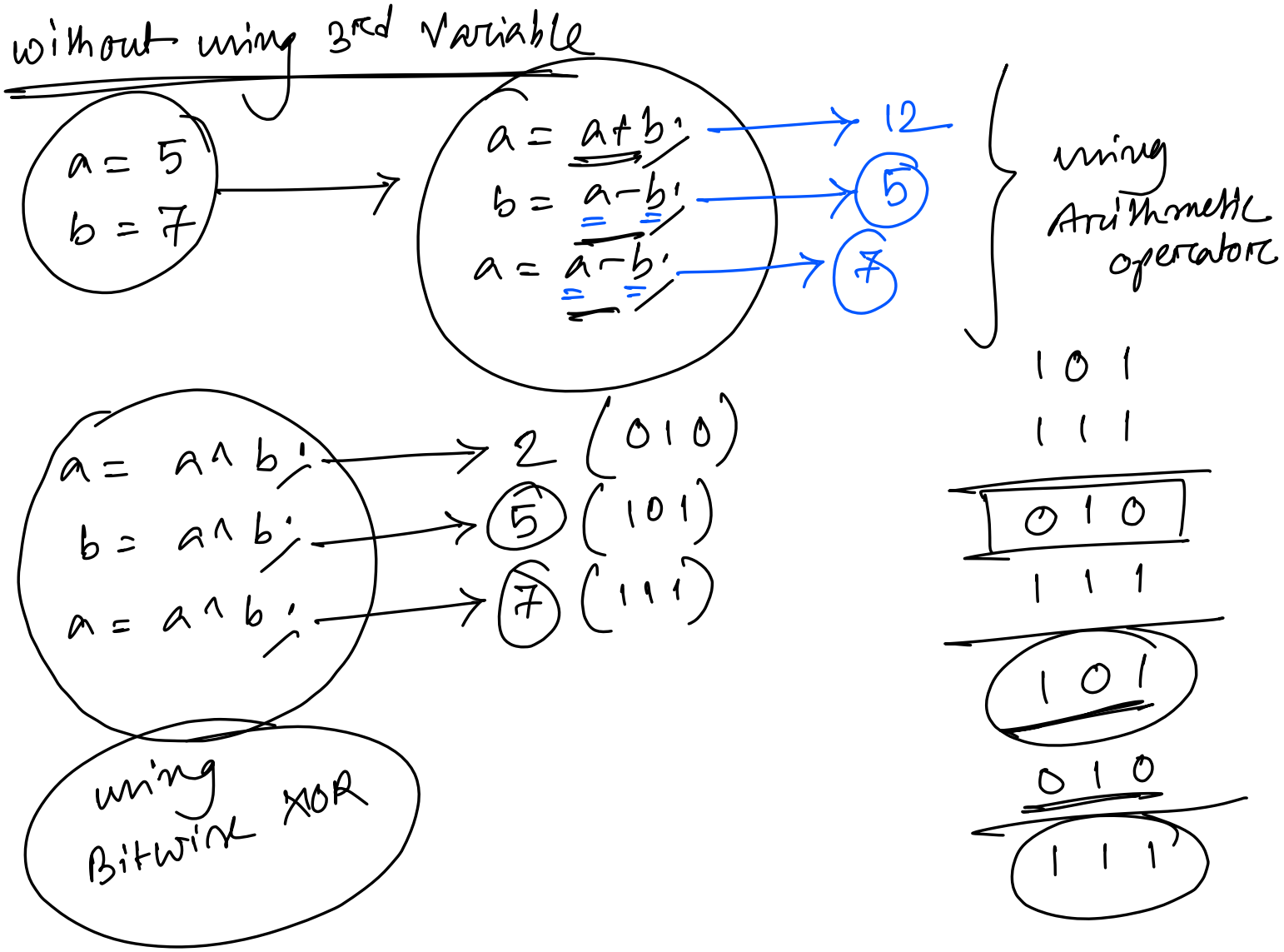
a = 10, b = 15

A = 15 | b = 10

1. using 3rd variable.
2. without using 3rd variable.




```
c = a;
a = b;
b = c;
```


without using 3rd Variable
a = 5
b = 7
a = a+b;
b = a-b;
a = a-b;
12
5
7
using Arithmetic operators
a = a^b;
b = a^b;
a = a^b;
2 (010)
5 (101)
7 (111)
using Bitwise XOR
1 0 1
1 1 1
0 1 0
1 1 1
1 0 1
0 1 0
1 1 1