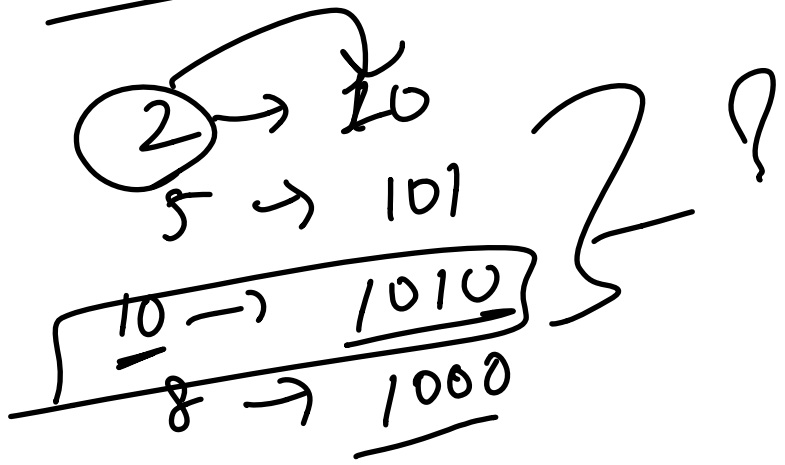
Decimals & Binary

① Decimal to Binary



① n = 10

→ / by 2

→ store remainder in answer

→ repeat above 2 step until n != 0

→ reverse answer

n = 10 → 1010

divisor = 2

Division	Rem
10/2 → 5	0
5/2 → 2	1
2/2 → 1	0
1/2 → 0	1

Rukyara

2 | 10 | 5
10
0

2 | 2 | 1
2
0

0 1 0 1

↓ reverse

1 0 1 0

answer

(2) n = 7 → 111

Division	Rem
7/2 → 3	1
3/2 → 1	1
1/2 → 0	1

Rukna

answer

1 1 1

↓ reverse

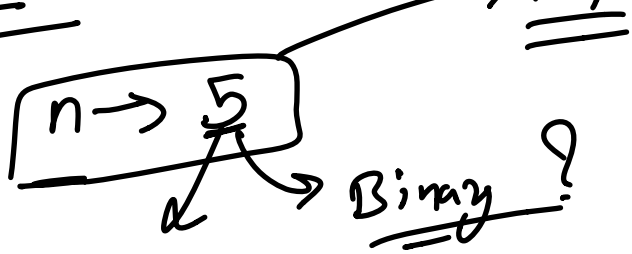
1 1 1

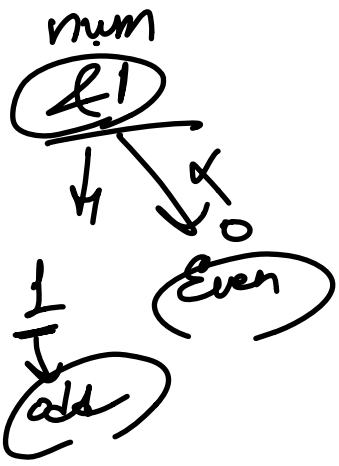
→ 1 1 1

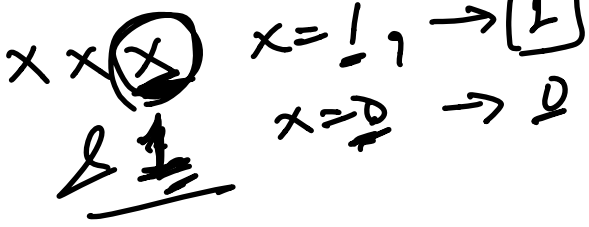
$2^2 + 2^1 + 2^0$

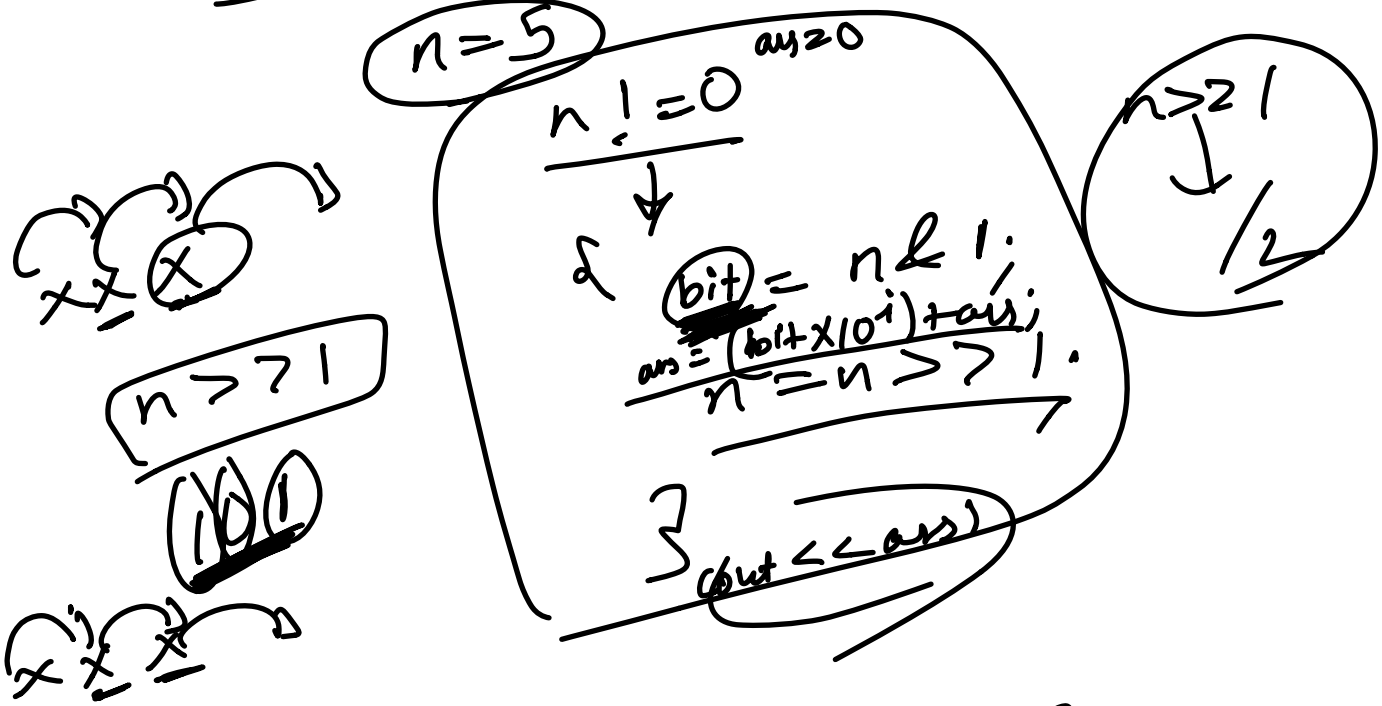
$4 + 2 + 1 = 7$

(2)nd Approach









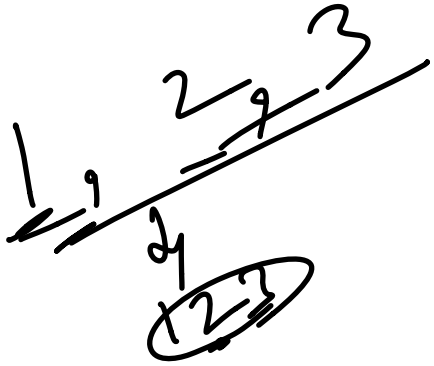

```
while (n!=0)
{
    bit = n&1;
    n = n>>1;
}
```

answer = ? 101

5 → 101 → 1 → 01 → 101

```
answer=0;
answer = (10^0 * digit) + answer
       = 10^0 * 1 + 0
       = 1 * 1 + 0 = 1
answer = (10^1 * digit) + answer
       = (10 * 0) + 1
```

20+1 = 1



ans = (10² × digit) + ans

= (100 × 1) + 1

= 100 + 1 = 101

ans = 0

ans = digit × 10⁰ + ans

= 1 × 10⁰ + ans

= 1

ans = (digit × 10¹) + ans

= 2 × 10 + 1

= 21

ans = (digit × 10²) + ans

= 3 × 100 + 21

= 321

ans = (digit × 10ⁱ) + ans

reverse

1 2 3

ans = 0

ans = (ans × 10) + digit

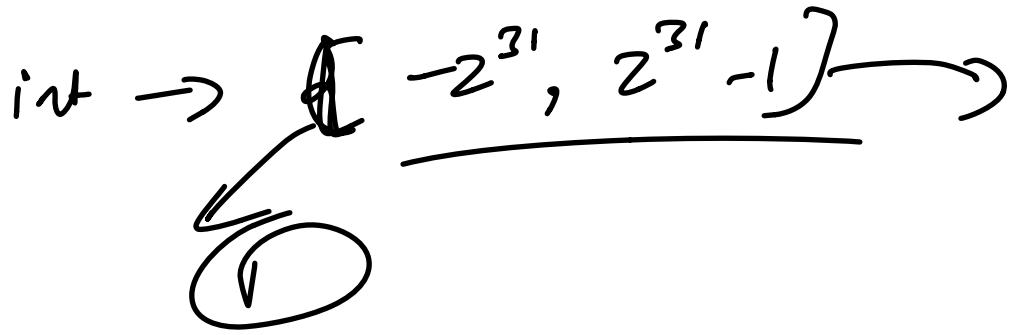
= (0 × 10) + 1 = 1

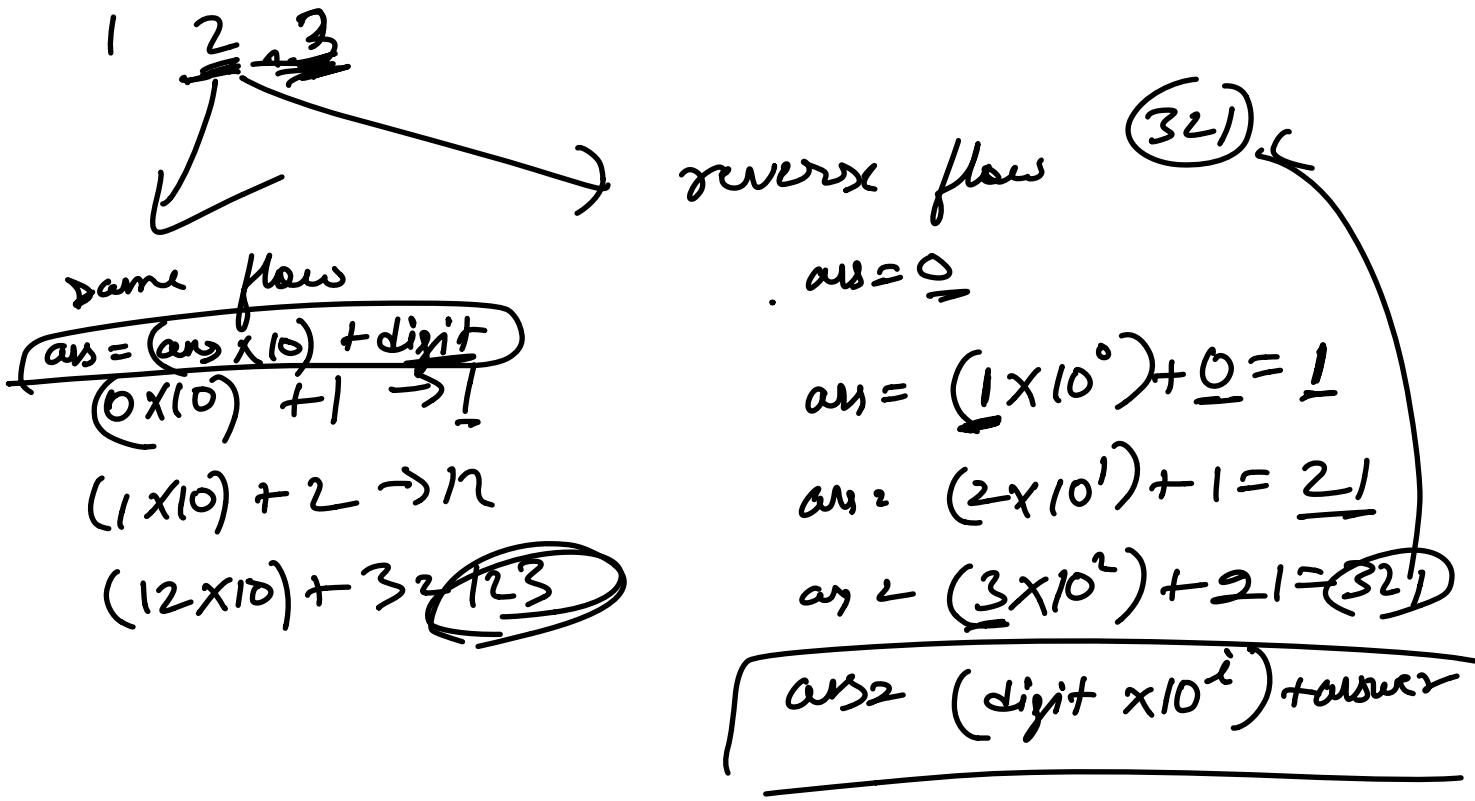
ans = (ans × 10) + digit

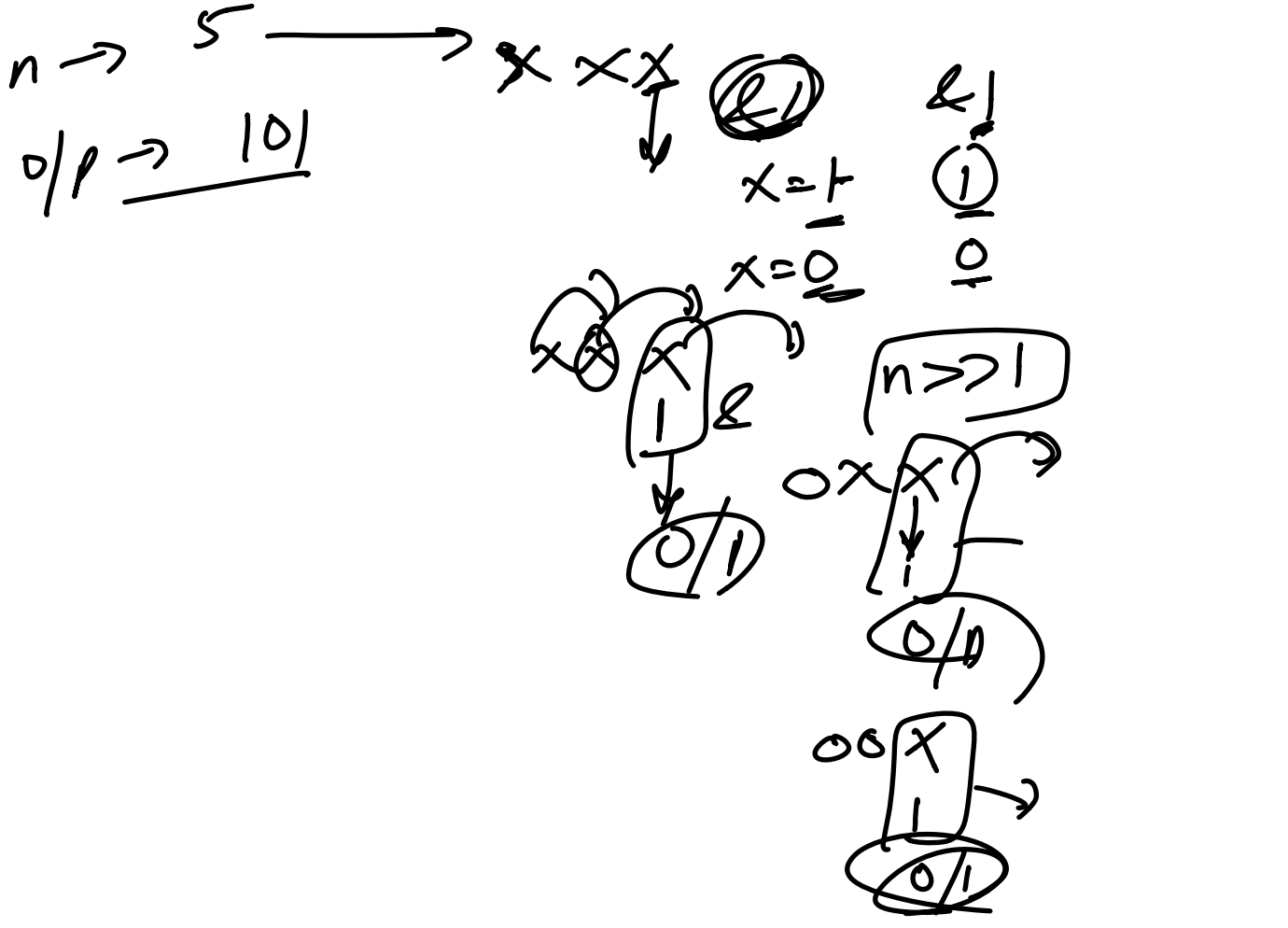
= (1 × 10) + 2 = 12

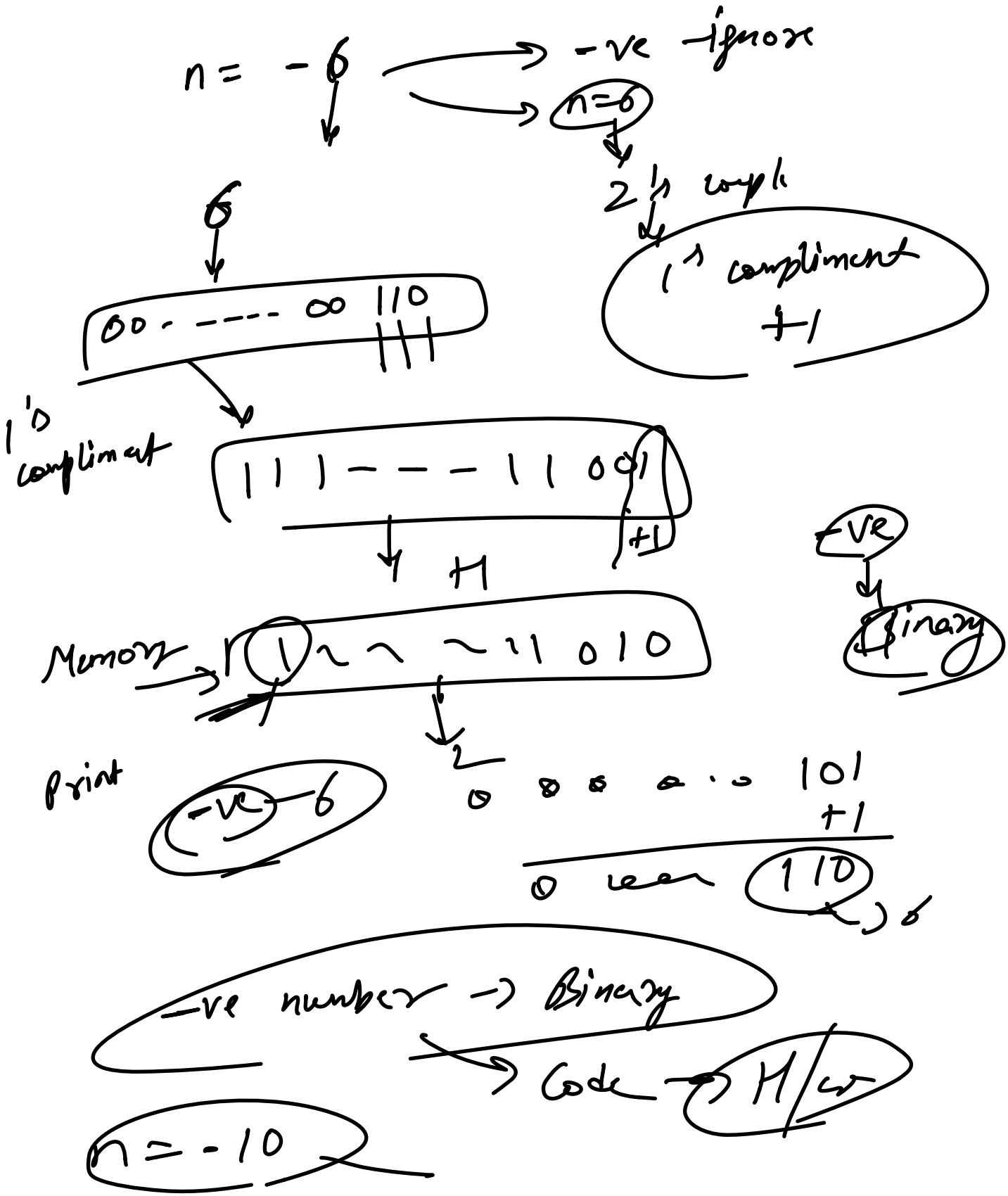
ans = (ans × 10) + digit

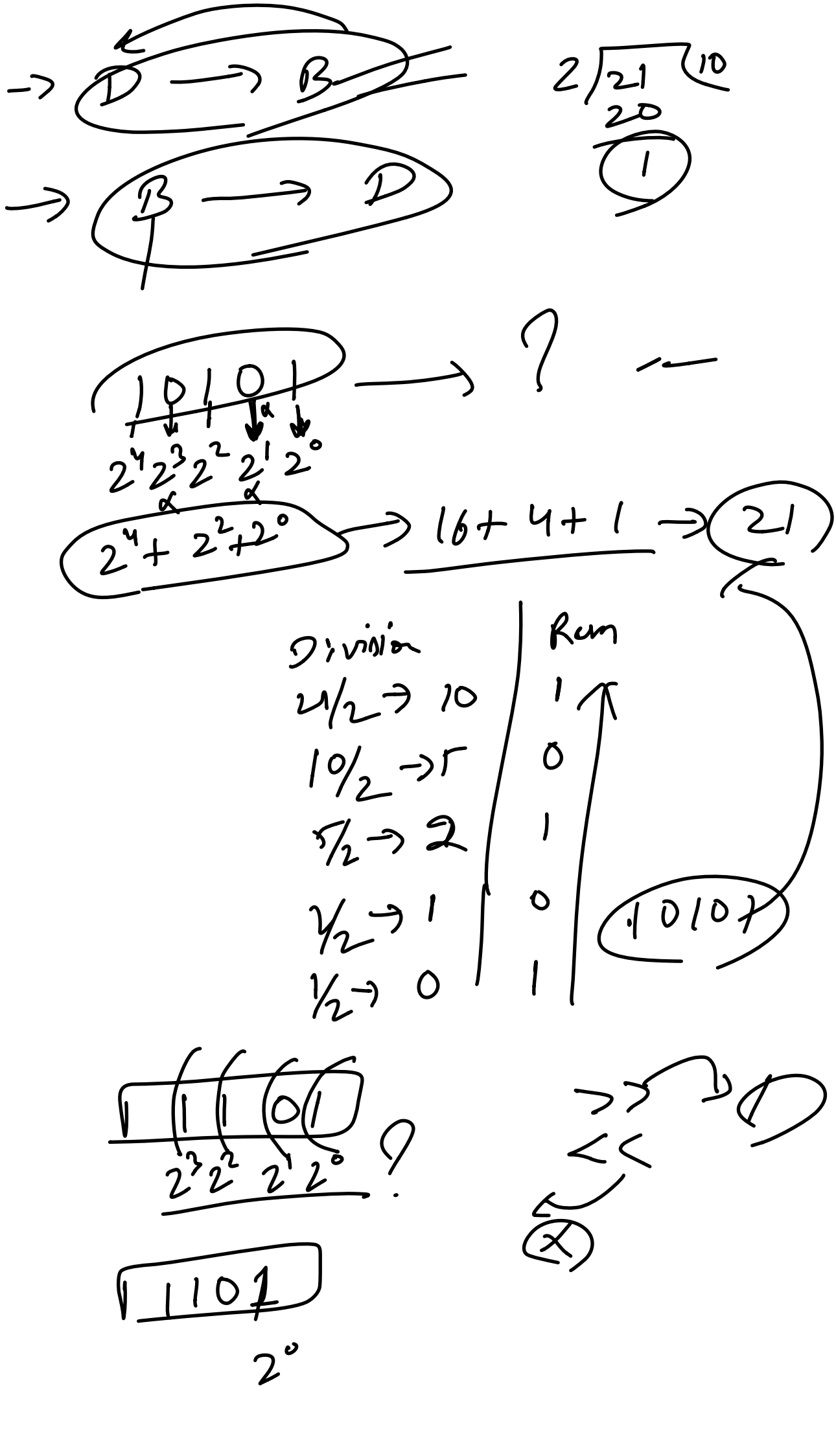
= (12 × 10) + 3 = 123

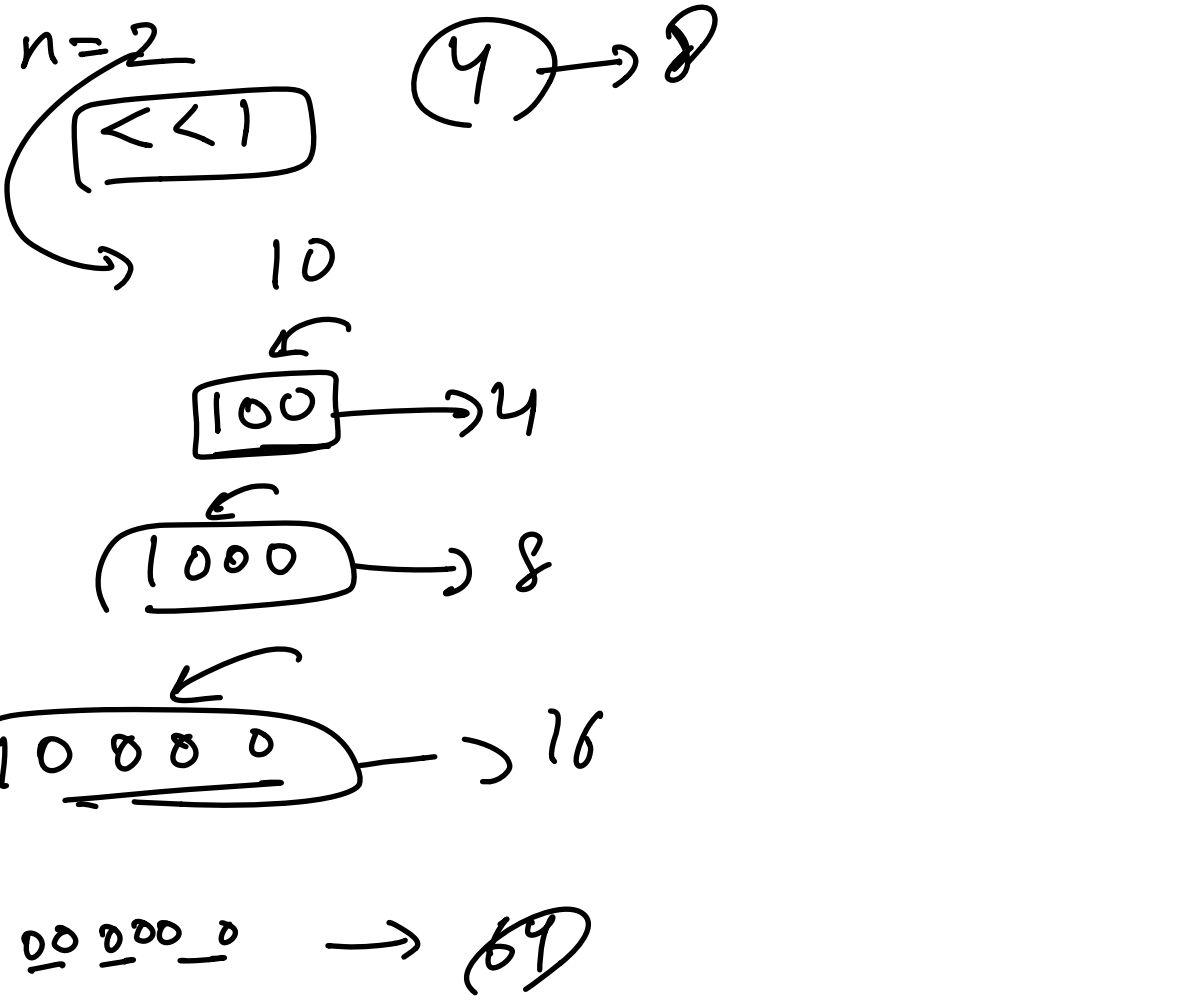

int → [-2^{31}, $2^{31}-1$] →
1

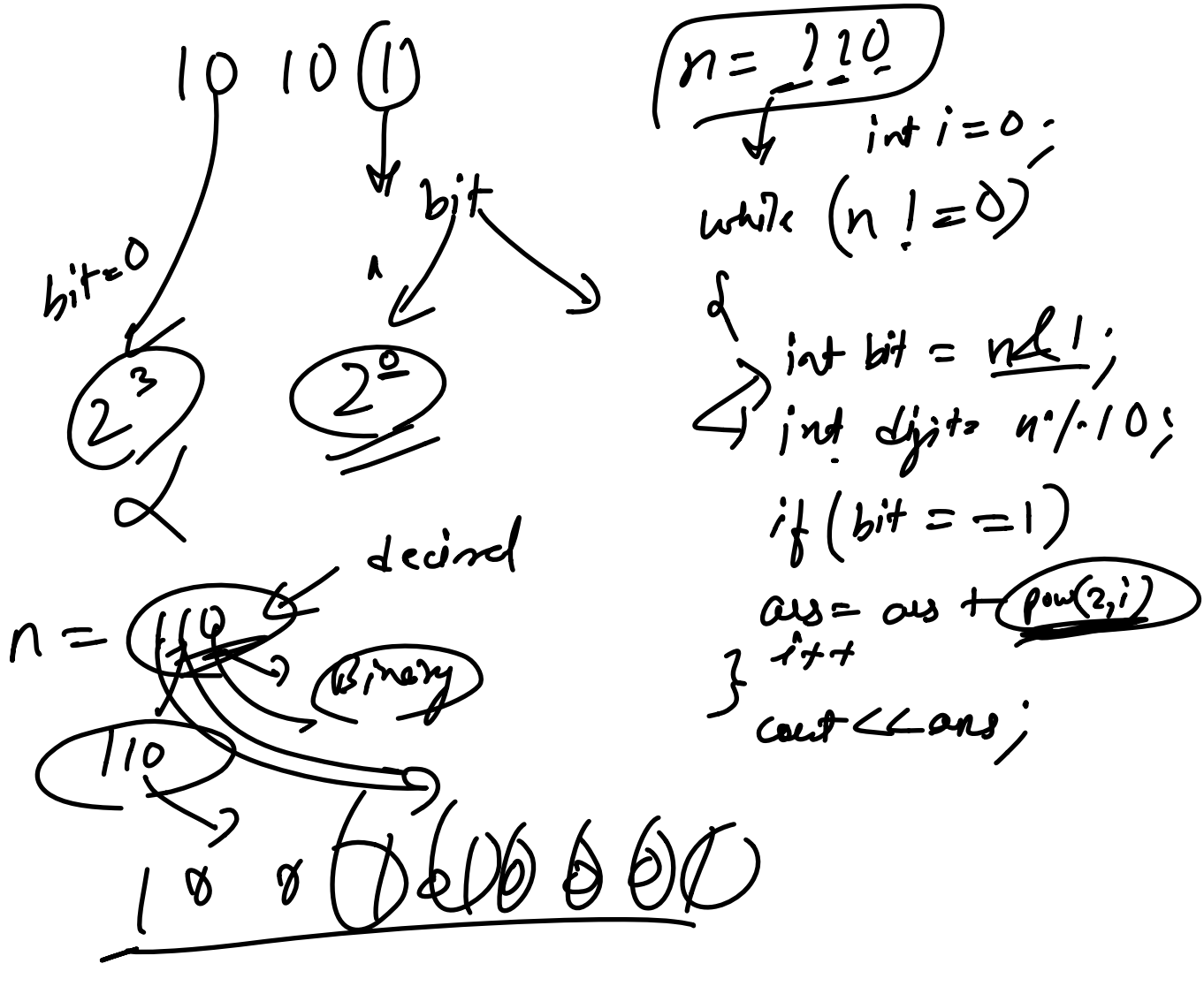

1 2 3
same flow
ans = (ans x 10) + digit
(0x10) +1 → 1
(1x10) + 2 → 12
(12x10) + 3 → 123
reverse flow
321
ans = 0
ans = (1x10⁰) + 0 = 1
ans = (2x10¹) + 1 = 21
ans = (3x10²) + 21 = 321
ans = (digit x10ⁱ) + answer

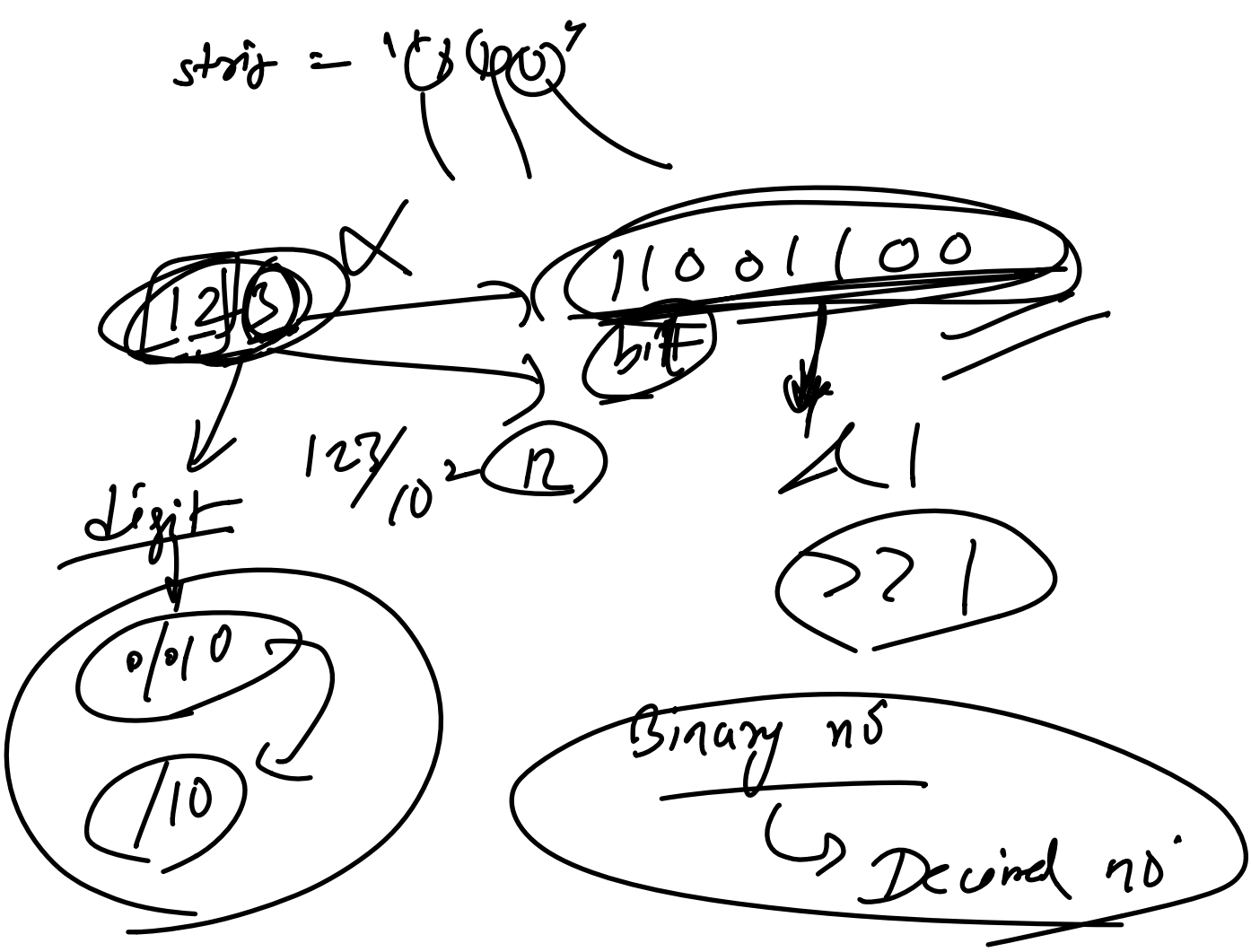

n → 5
o/p → 101
x x x
x=1
x=0
n>>1
0/1

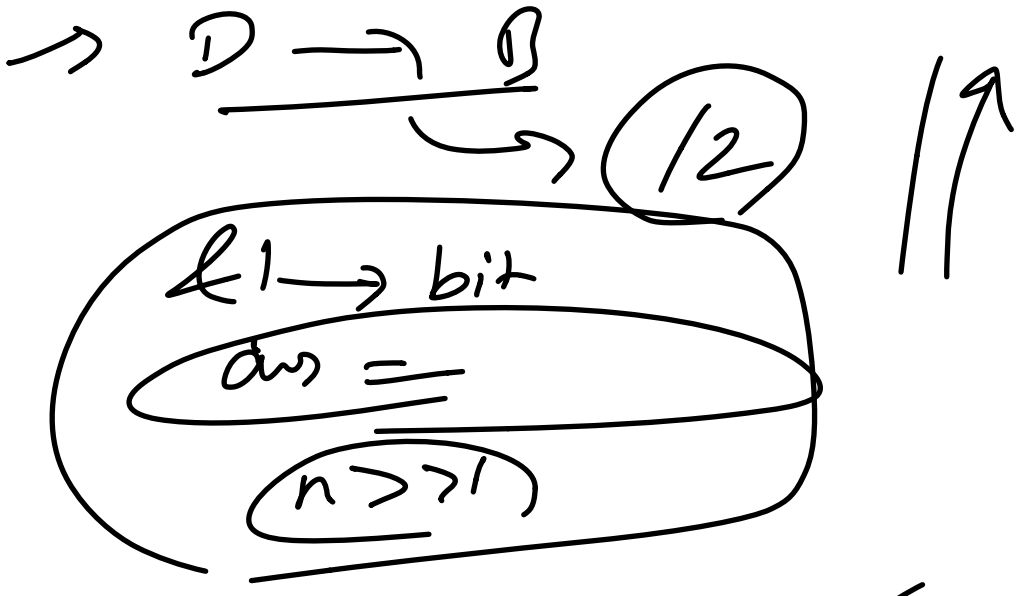

n = -6
-ve ignore
n=6
2's comp
6
00 ---- 00 110
1's compliment
+1
1's compliment
111 --- 11001
+1
+1
-ve
Binary
Memory
1 ~ ~ ~ 11010
Print
-ve 6
0 0 0 0 ... 101
+1
0 ... 110
-ve number → Binary
Code → H/w
n = -10

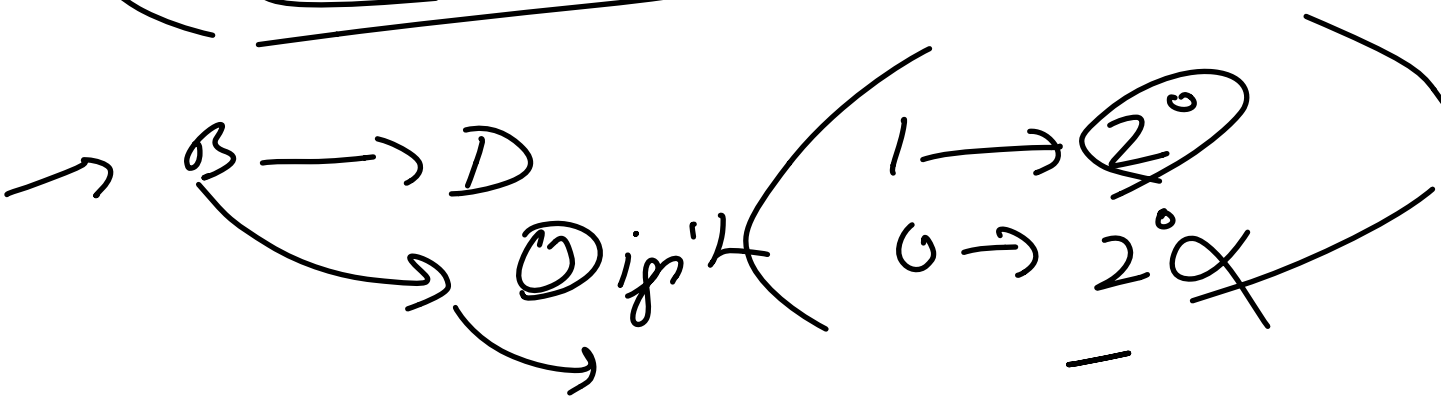

D → B
B → D
2 | 21 | 10
20
1
1 0 1 0 1
$2^4 2^3 2^2 2^1 2^0$
$2^4 + 2^2 + 2^0$
16 + 4 + 1 → 21
Division
Rem
21/2 → 10 1
10/2 → 5 0
5/2 → 2 1
2/2 → 1 0
1/2 → 0 1
10101
1 1 0 1
$2^3 2^2 2^1 2^0$
1101
2^0


n=2
<<1
4 → 8
10
100 → 4
1000 → 8
10000 → 16
100000 → 69


10 10 (1)
bit
bit=0
2^3
2^0
n = 110
decimal
Binary
110
1 0 0 0 0 0 0
n = 110
int i=0;
while (n != 0)
{
int bit = n%1;
int digit = n/10;
if (bit == 1)
ans = ans + pow(2,i)
i++
}
cout << ans;


str = '123'
123
bit
digit
Binary no
Decimal no


bit
n >> 1


Digit
1 → 2^0
0 → 2^0