Chapter-6

Binary Number System

Binary Number System

0/1

Base 2

Math : Decimal Number System

0-9 // 10 Digits

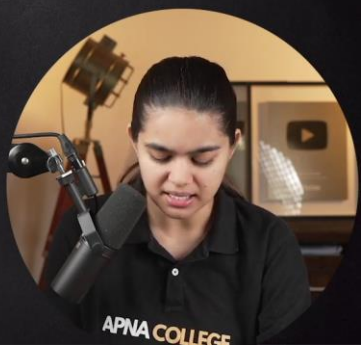
Base 10

Hexadecimal

Base 16

Octal

Base 8



Decimal to Binary

2 to 10

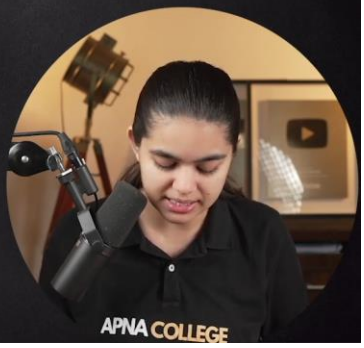
2	42	0
2	21	1
2	10	0
2	5	1
2	2	0
2	1	1
	0	

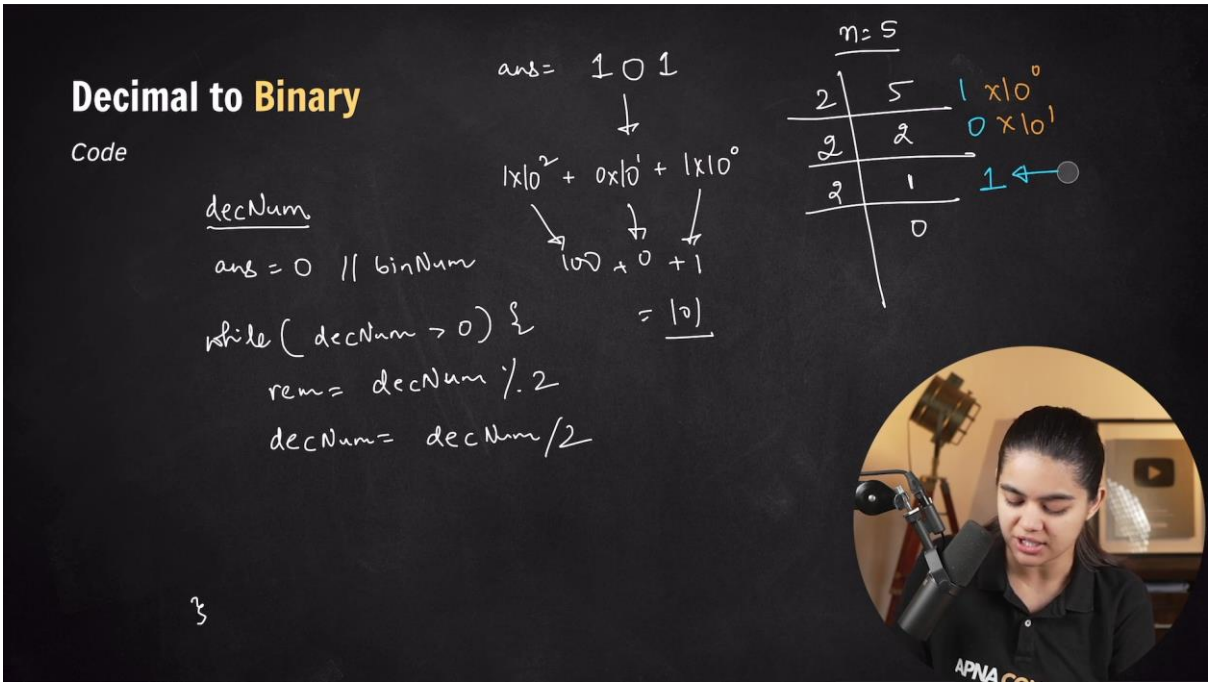
$(42)_{10} = (101010)_2$

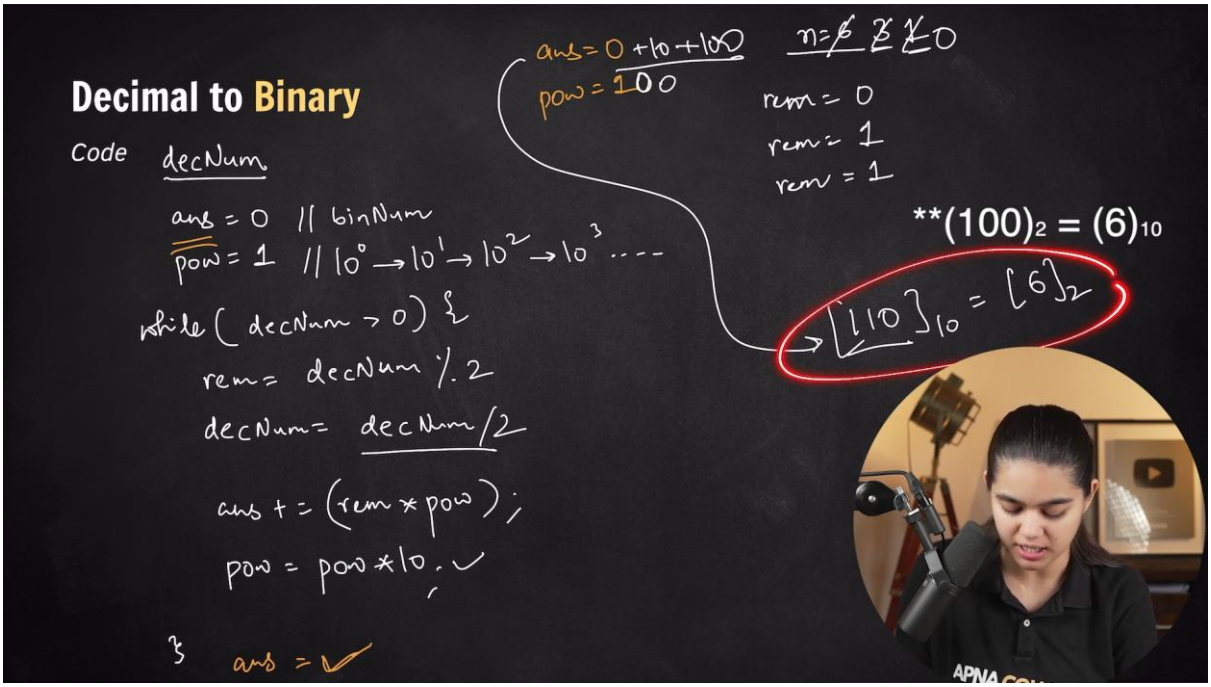
2	50	0
2	25	1
2	12	0
2	6	0
2	3	1
2	1	1
	0	

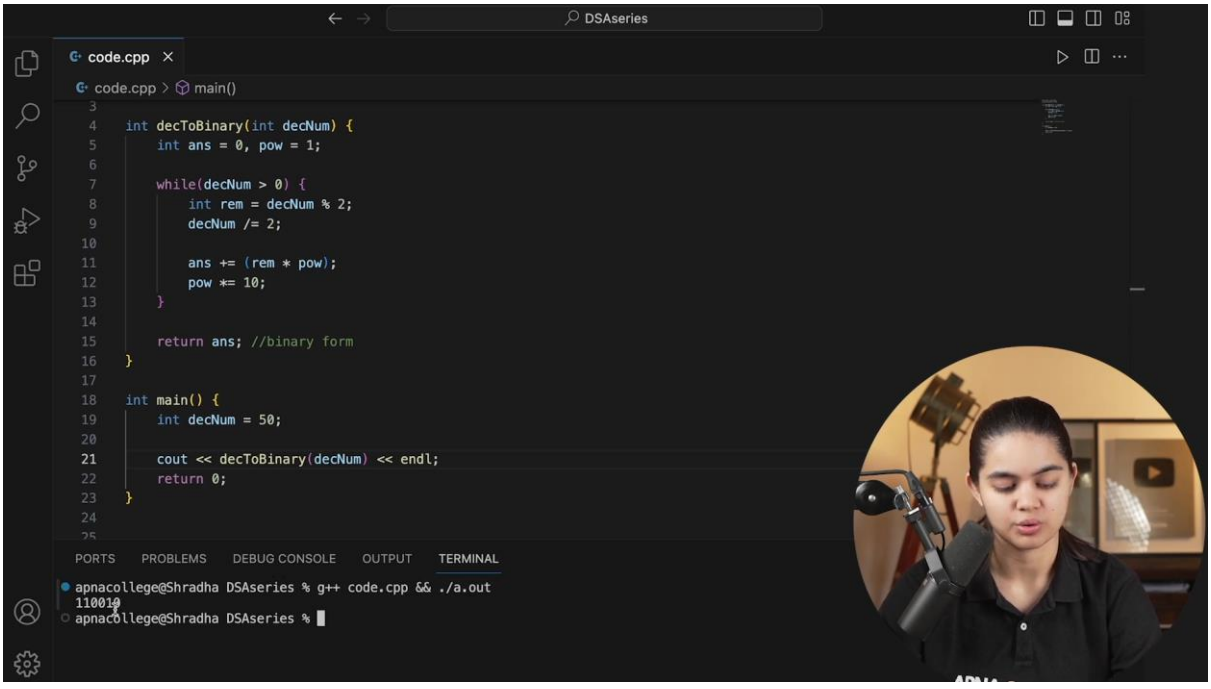
$(110010)_2$

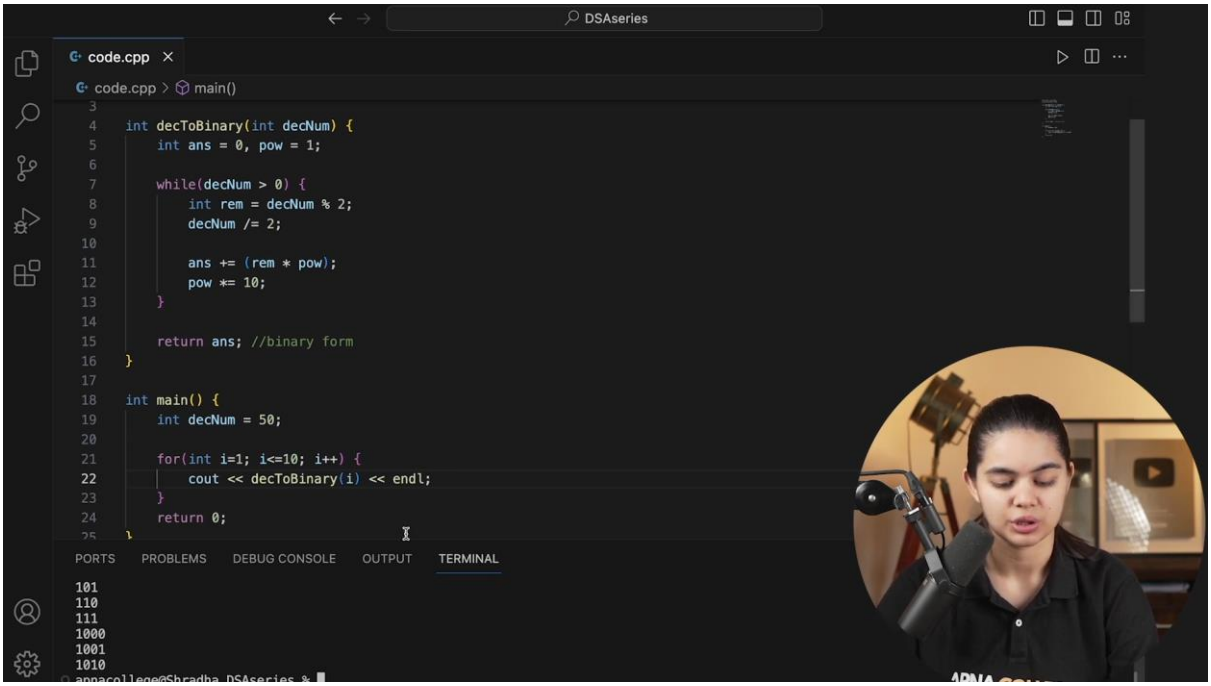
$= (50)_{10}$

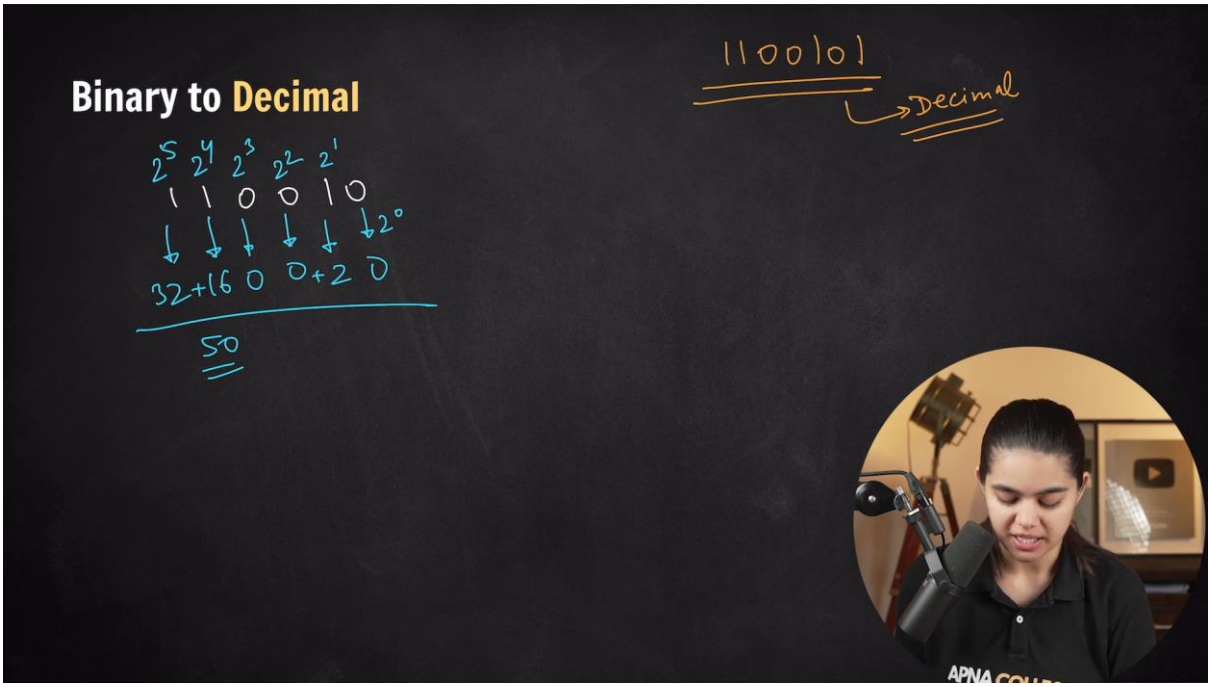


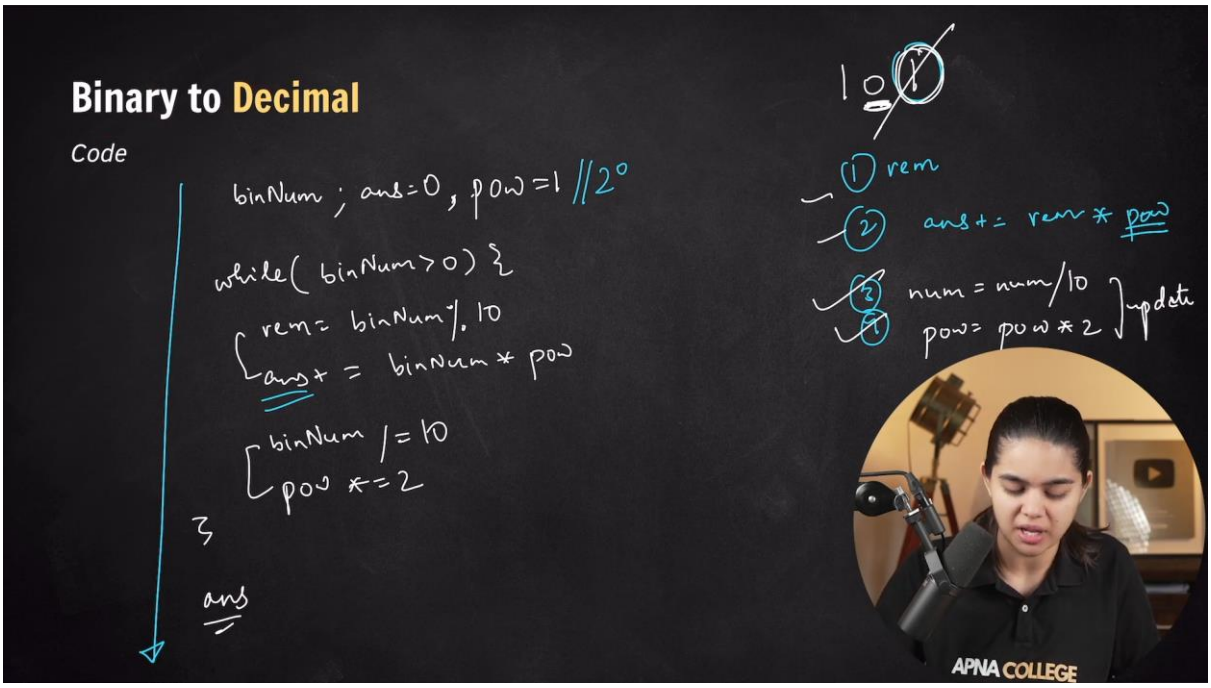

Decimal to Binary
Code
decNum
ans = 0 // binNum
while (decNum > 0) {
rem = decNum % 2
decNum = decNum /2
}
ans = 101
1x10^2 + 0x10^1 + 1x10^0
100 + 0 + 1
= 101
n = 5
2 | 5 | 1 x10^0
2 | 2 | 0 x10^1
2 | 1 | 1
0


Decimal to Binary
Code
decNum
ans = 0 // binNum
pow = 1 // 10^0 → 10^1 → 10^2 → 10^3 ....
while (decNum > 0) {
rem = decNum % 2
decNum = decNum /2
ans += (rem * pow);
pow = pow * 10;
}
ans =
ans = 0 + 10 + 100
pow = 100
n = 6 3 1 0
rem = 0
rem = 1
rem = 1
**(100)2 = (6)10
[110]10 = [6]2


DSAseries
code.cpp
code.cpp > main()
int decToBinary(int decNum) {
int ans = 0, pow = 1;
while(decNum > 0) {
int rem = decNum % 2;
decNum /= 2;
ans += (rem * pow);
pow *= 10;
}
return ans; //binary form
}
int main() {
int decNum = 50;
cout << decToBinary(decNum) << endl;
return 0;
}
PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
110010
apnacollege@Shradha DSAseries %


DSAseries
code.cpp
code.cpp > main()
int decToBinary(int decNum) {
int ans = 0, pow = 1;
while(decNum > 0) {
int rem = decNum % 2;
decNum /= 2;
ans += (rem * pow);
pow *= 10;
}
return ans; //binary form
}
int main() {
int decNum = 50;
for(int i=1; i<=10; i++) {
cout << decToBinary(i) << endl;
}
return 0;
}
PORTS
PROBLEMS
DEBUG CONSOLE
OUTPUT
TERMINAL
101
110
111
1000
1001
1010


Binary to Decimal
1100101
Decimal
2^5 2^4 2^3 2^2 2^1
1 1 0 0 1 0
2^0
32+16 0 0+2 0
50


Binary to Decimal
Code
binNum ; ans=0, pow=1 //2^0
while(binNum>0) {
rem= binNum%10
ans+ = binNum * pow
binNum /=10
pow *=2
}
ans
1 0 1
1 rem
2 ans+= rem * pow
num = num/10
pow= pow*2
update
APNA COLLEGE

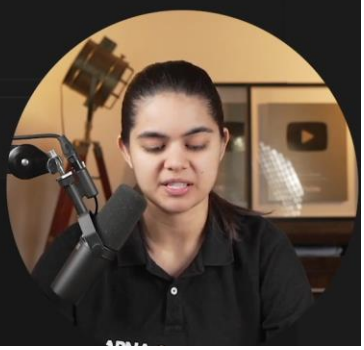
DSAseries

code.cpp

```cpp
int binToDecimal(int binNum) {
    int ans = 0, pow = 1;

    while(binNum > 0) {
        int rem = binNum % 10;
        ans += rem * pow;

        binNum /= 10;
        pow *= 2;
    }

    return ans; //decimal form
}

int main() {
    cout << binToDecimal(101) << endl; //5
    return 0;
}
```

PORTS PROBLEMS DEBUG CONSOLE OUTPUT TERMINAL

```
apnacollege@Shradha DSAseries % g++ code.cpp && ./a.out
5
apnacollege@Shradha DSAseries %
```

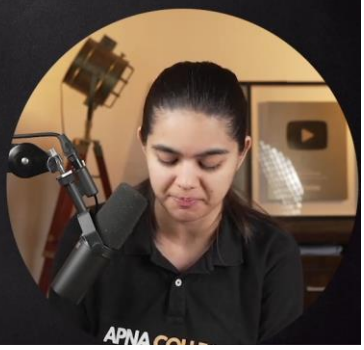


Common Numbers

25 − 16 = 9 − 8 = 1 − 1 = 0

1 1 0 0 1
16 8 4 2 1
✓ ✓ x x ✓

0 → 00
1 → 01
2 → 10
3 → 11
4 → 100
5 → 101
6 → 110
7 → 111
8 → 1000
9 → 1001
10 → 1010



Common Numbers

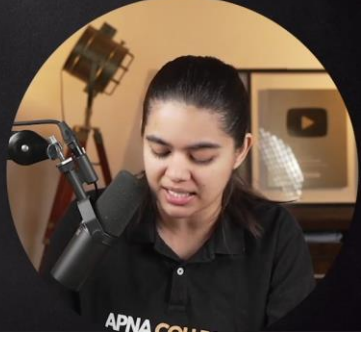
36 − 32 = 4 − 4 = 0

1 0 0 1 0 0
32 16 8 4 2 1
✓ x x ✓ x x

1

37
41
19
Bin

0 → 00
1 → 01
2 → 10
3 → 11
4 → 100
5 → 101
6 → 110
7 → 111
8 → 1000
9 → 1001
10 → 1010



Common Numbers

0 + 0 = 0

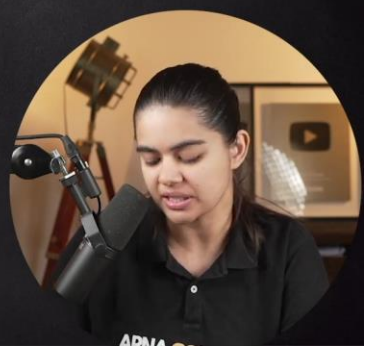
0 + 1 = 1

1 + 0 = 1

1 + 1 = 10

0 → 00
1 → 01
2 → 10
3 → 11
4 → 100
5 → 101
6 → 110
7 → 111
8 → 1000
9 → 1001
10 → 1010

1 + 1 = 10



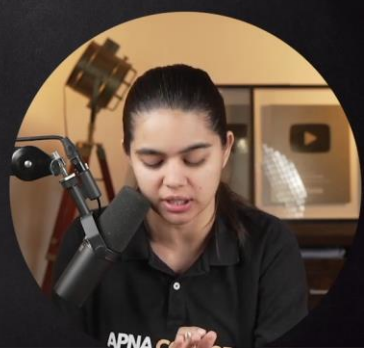
Two's complement

int n = -10

(-10)$_{10}$ = (10110)$_2$

1. Binary Form convert
2. Prefix with a 0
3. 1's complement
   - 0 → 1
   - 1 → 0
4. add +1

1. 01010
2. 10101 (+1)
3. 10110



Two's complement

-ve

10110

1. 01001 (+1)
2. 01010

(1010) → (10)$_{10}$

(-10)

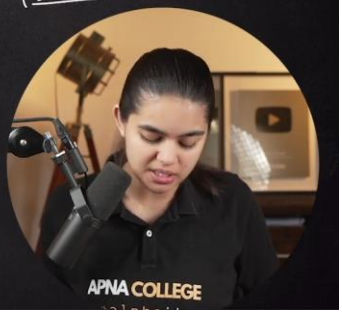
Bin → Dec

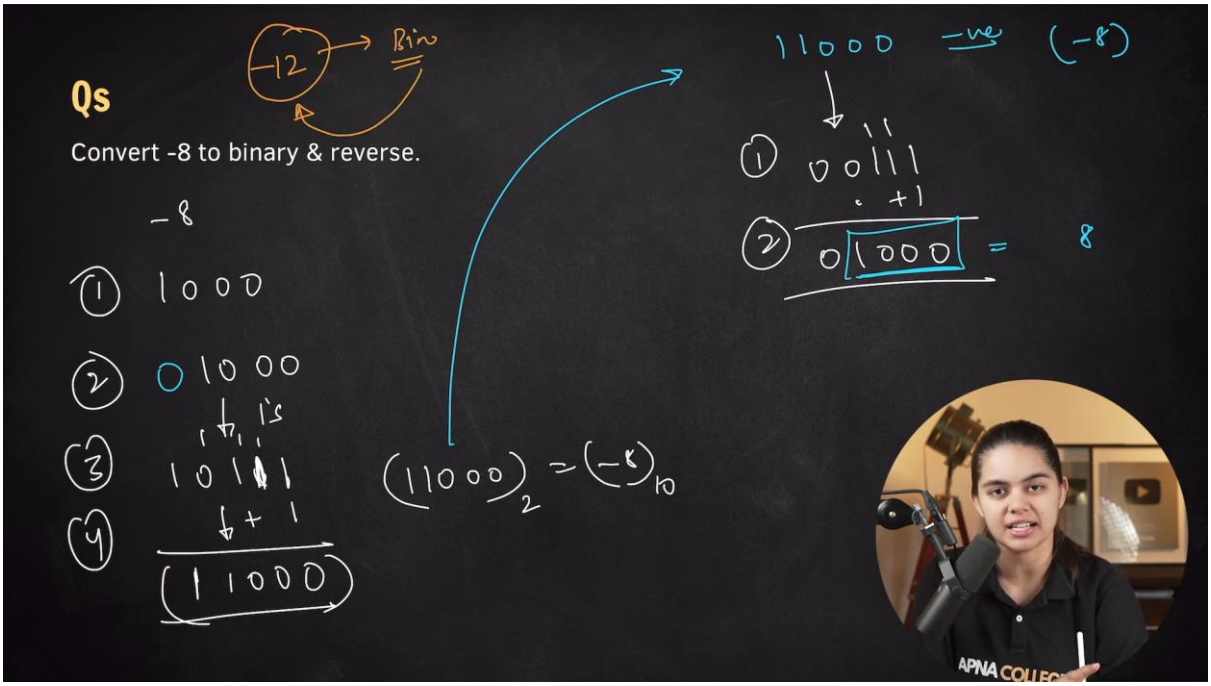
2's comp

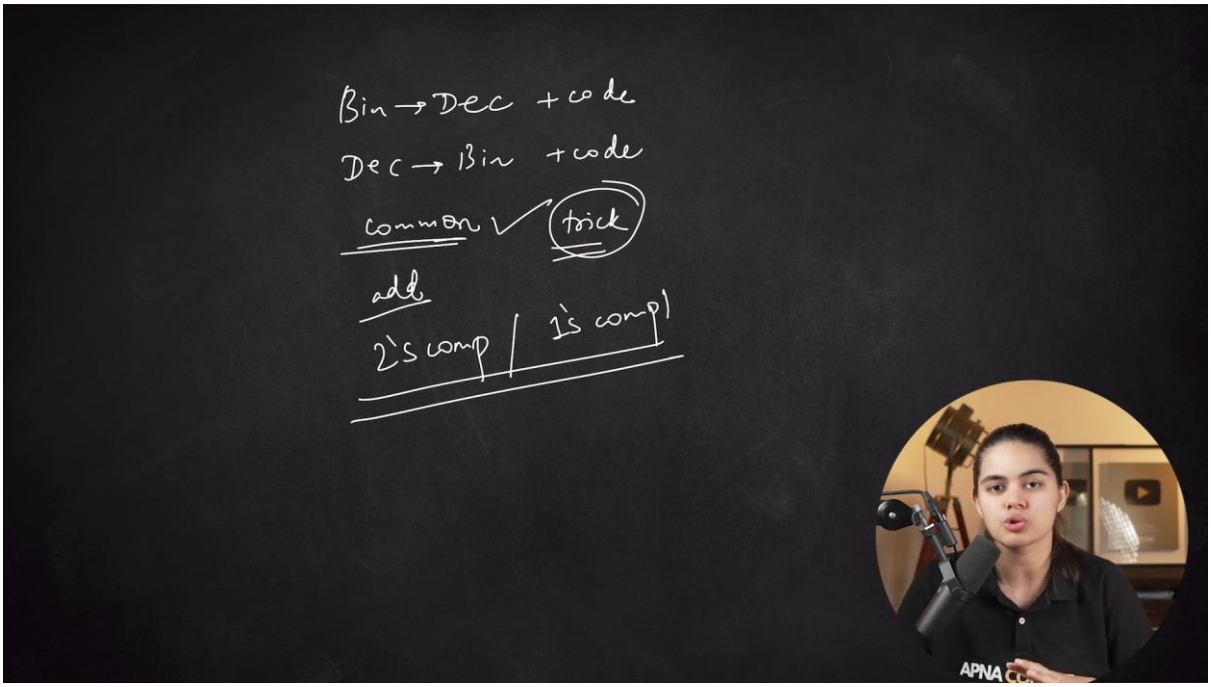
1. 1's comp
2. +1

1. Binary Form convert
2. Prefix with a 0
3. 1's complement
   - 0 → 1
   - 1 → 0
4. add +1

Dec → Bin







Homework Solutions:

0001

0010

0011

0100

0101

0110

0111

1000

1001

1010