Date
Page 1

BIT MANIPULATION

0000 → 0
0001 → 1
0010 → 2
0011 → 3
0100 → 4

1. Get bit at specific position

$n = 0101 = 5$ (3 2 1 0)

Suppose we need to get bit at position, i=2

$2 = 0010$ (3 2 1 0)

$1 << i = 0100$

$0001 \rightarrow 0010 \rightarrow 0100$

$0101 \;\&\; 0100 = 0100$

if $n \;\&\; (1 << i) != 0$, then bit is 1

if bit at given position is not zero (0) then bit is 1

Code:

```
int getBit(int n, int pos){
    return ((n & (1 << pos)) != 0)
}

getBit(5,2); ⇒ 1
```

2. Set bit

make specified bit to '1'

unset means 0

n = 0101

Suppose we need to set bit at position, i=1

1 << i = 0010

0101 | 0010 = 0111

or

Code :-

```
int setBit (int n , int pos) {
    return (n | (1 << pos));
}
```

setBit (5, 1); ⇒ 7 0111

3. Clear Bit — make bit to '0'

n = 0101 = 5

suppose we need to clear bit at position, i = 2

1 << i = 0100

~0100 = 1011 (once complement) flip the bit

0101 & 1011 = 0001

Code:-

```
int clearBit (int n, int pos) {
    int mask = ~(1 << pos);
    return (n & mask);
}

clearBit (5, 2);  → 1
```

Date
Page 4

4. Update bit

n = 0101

Suppose we need to update bit at position, i = 1 to 1

1 << i = 0010

~0010 = 1101

0101 & 1101 = 0101

clear Bit

1 << i = 0010

0101 | 0010 = 0111

= 7

set Bit

Code:

```
int updateBit(int n, int pos, int value){
    int mask = ~(1<< pos);
    n = n & mask;
    return (n | (value << pos));
}
```

updateBit(5, 1, 1); → 7