Q1) Row wise & col wise matrix

Adv DSA arrays 2 first ques

1 2 3
4 5 6
7 8 9

k = 2

start at
10:35

Start at i = 0 j = n-1

i, j set i×1009 + j

assume 1 index

i = 17 → 18

j = 21 → 22

1 2 2 3
4 5 6 7
8 9 10 11

k = 2

2 1 0

A = 1 1 1

conv A = 0 0 0

Q2 A, B, C.

1) unset C bits from right.

2) Restore B bits from right to original value

Eg1 A = 12 B = 2 C = 3

3 2 1 0

1 ~~1~~ 0 0

0

$\Rightarrow$ ans = 8

Eg2 A = 7 B = 1 C = 3

2 1 0

~~1~~ ~~1~~ ~~1~~

0 0 ~~1~~ 1

ans = 1

31 ..... C C-1 ... B+1 B B-1 ..... 2 1 0

Let A: 0 0 0 0 ~~0~~ ~~0~~ ~~0~~ ~~0~~ ~~0~~

0g...0g 0g 0g 0g 0g 0g

In A set [B, C-1] = 0

```
for (bit = B ; bit <= C-1 ; bit ++) {
        unset (A, bit)
}
```

```
unset ( int A , int bit ) {
        if ( A & 1<<bit != 0) {
                A = A ^ 1<<bit
        }
}
```

Approach 2

```
orginal_a = A
for ( bit =0 ; bit <C ; bit++) {
        unset (A, bit)
}
```

Step 1

```
for ( bit = 0; bit <B ; bit ++) {
        if ( orginal_a & 1<<bit != 0) {
                set (A, bit)
}
```

Q3 Max possible subarray sum with decreasing weight

Eg1 3 2 1 ans = 6

Eg2 3 3 5 0 1 ans = 5

have we seen some thing like this before ?

Yes

Kadane's Algorithm

⇒ Keep taking until love >, 0

Here we have same idea but different condition.

Keep taking while elems are decreasing.

	0	1	2	3	4	5	6
	7	9	8	4	1	3	10
sum =	7	9	17	21	22	3	10
ans =	7	9	17	21	22	22	22

Code

```
int solve ( int [] arr, int N) {
    int ans = arr[0]
    int cur_sum = arr[0]
    for ( i=1 ; i<n ; i++ ){
        if (a[i] < a[i-1]) {
            cur_sum += a[i] ; ans = max (ans,
                                         cur_sum)
        }
        else {
            cur_sum = a[i]
            ans = max (ans, cur_sum )
        }
    }

    return ans
}
```

Repeat / Reattempt

Sat / Sun