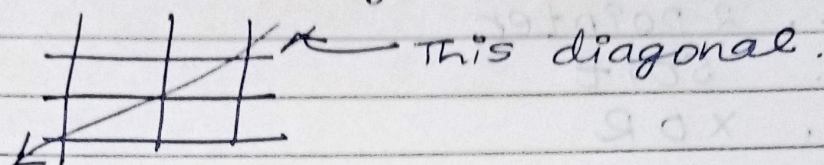


Doubt Week-4

Diagonal Printing



Code:

```

int i = 0, j = 2;
for (int j; j >= 0; j--) {
    cout << a[i++] [j]
}

```

Swap implement $\Rightarrow (a, b)$ 1st Method

- ① Temp variable
- ② $(+, -) \rightarrow$ Arithmetic
- ③ XOR

② int a, int b

a = 2, b = 3

a = a + b \rightarrow 1b = a - b \rightarrow 2a = a - b \rightarrow 3

a = 3, b = 2

③ int a, int b

a = a ^ b \rightarrow 1b = a ^ b \rightarrow 2a = a ^ b \rightarrow 3

E 11

12-10-2021

Arrays

1. sliding window
2. 2 pointer
3. Sort
4. XOR
5. ADD all

Rotate Array using Reverse Array

1 | 2 | 3 | 4 | 5 | 6 | 7

IF $K=3$,

① 7 | 6 | 5 | 4 | 3 | 2 | 1

② 5 | 6 | 7 | 4 | 3 | 2 | 1

 $0 \rightarrow K-1$

Reverse

③ 5 | 6 | 7 | 1 | 2 | 3 | 4 |

 $K \rightarrow n-1$

Code:

```

void rotate(vector<int> &nums, int k) {
    int n = nums.size();
    k = k % n;
    reverse(nums.begin(), nums.end());
    reverse(nums.begin(), nums.begin() + k);
    reverse(nums.begin() + k, nums.end());
}

```


Find missing Number

9	6	4	2	3	5	7	0	1
0	1	2	3	4	5	6	7	8

XOR with index

Rearrange Array Elements by sign.
Approach NO 1.

Will use 2 extra spaces for segregating -ve numbers and +ve numbers.

After segregating, we will push one by one in ans vector.

Code:

```
vector<int> rearrangeArray(vector<int> &nums) {
    int n = nums.size();
    vector<int> ans(n);
    int i = 0;
    int j = 1;
    for (auto it : nums) {
        if (it >= 0) {
            ans[i] = it;
            i += 2;
        }
        else {
            ans[j] = it;
            j += 2;
        }
    }
    return ans;
}
```

};