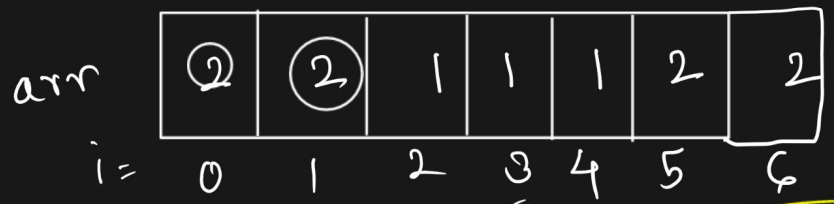
Day - 6 DSA

* Leetcode Pbm No :- 169 - Majority Element



count = 0

result = 2

major = 2

i=0 for (2) 2 = result ✗ 1

i=1 for 2 2 = result ✗ 2

i=2 for 1 1 ≠ result ✗ 1

i=3 for 1 1 ≠ result ✗ 0

i=4 for 1 1 = 1 ✓ 1

i=5 for 2 2 ≠ 1 ✗ 0

i=6 for 2 (2) = result ✓ 1

i=7 stop

major = 2

result = 1

result = 1

result = 2

result = 2

```
int majorElement (int arr[], int n)
{   int count = 0 , int result = 0

    for(int = 0 ; i<n ; i++)
    {
        if ( count == 0)
        {
            result = arr[i];
        }
        count += (arr[i] == result) ?
                  1 : -1 ;
    }

    return result;
}
```

[2, 2, 2, 1, 1, 1]

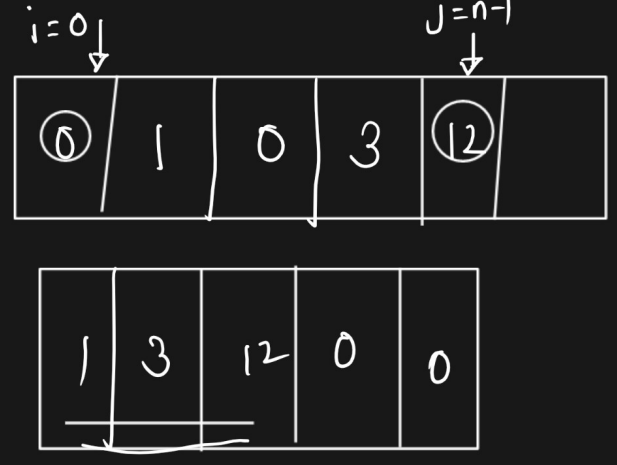
0 1 2 3 4 5

[2, 2, 1, 1, 2, 1]

[1, 1, 2, 2, 1, 2]

		count	reset	arr[i]==r	count
i=0	⇒	0	2	2=2 ✓	0+1=1
i=1	⇒	1	2	2=2 ✓	1+1=2
i=2	⇒	2	2	1≠2	2-1=1
i=3	⇒	1	2	1≠2	1-1=0
i=4	⇒	0	2	2=2	0+1=1
i=5	⇒	1	(2)	1≠2	1-1=0

* LeetCode Pbm No :- 283 More Zero's :-



Swap

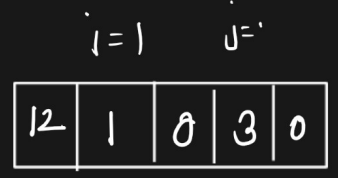
Cond^n :- if arr[i] == 0 ✓

&& arr[j] != 0 ✓

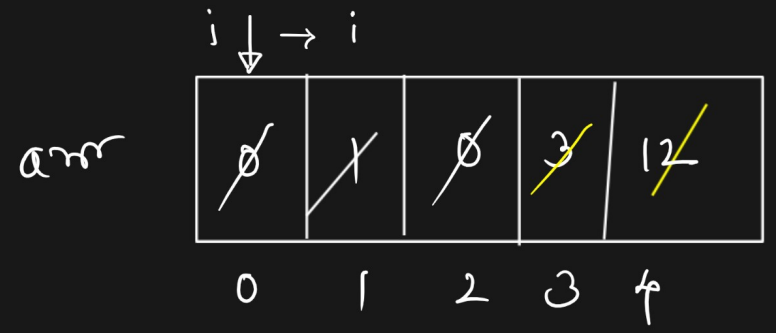
only then you can swap arr[i], arr[j]

arr[i] ≠ 0 → i++

or arr[j] = 0 → j--



} but relative order of ele doesn't maintained



zeroIndex = 0

i = 0 arr[i] ≠ 0 (1 ✓, x) arr[ ] → 0 1 2 3 4

i = 1 arr[i] ≠ 0 ✓ arr[1 _] 0 1 2 3 4 — zeroIndex ↑, ++

i = 2 arr[2] ≠ 0 (x) arr[1 _ _ _ _] 0 1 2 3 4 — zeroIndex ↑

i = 3 arr[3] ≠ 0 ✓ arr[1 3 _ _ _] 0 1 2 3 4 — zeroIndex ↑, ++

i = 4 arr[4] ≠ 0 ✓ arr[1, 3, 12, _, _] 0 1 2 3 4 — zeroIndex ↑

i == 4

for loop finished

```
for (int i = zeroIndex; i < arr.length; i++)
{
    arr[i] = 0;
}
```

```
int zeroinden = 0;
for(int i=0; i<n; i++)
{
    if(arr[i] != 0)
    {
        arr[zeroinden++] = arr[i];
    }
}
for(int i=zeroinden; i<n; i++)
{
    arr[i] = 0;
}
```

num zero

zero

0	1	0	3	12

zeroinden = 0

arr

i = 0 ; arr[0] != 0 ✓

0	1	0	3	12
0	1	2	3	4

i = 1 ; arr[1] != 0 ✓

1	1	0	3	12

zeroinden++

i = 2 ×

1	1	0	3	12

zi

i = 3 arr[3] != 0 ✓

1	3	0	3	12

zi++

i = 4 arr[4] != 0 ✓

1	3	12	3	12

i

~~for~~ zero index

i = 3

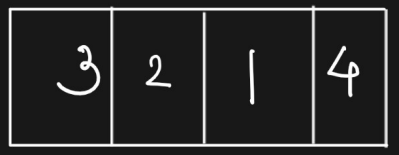
0	1	2	3	
1	3	12	0	12

zi

i = 4

1	3	12	0	0

H.W. i/p ⇒



value in betn 1 to n

n = 6

1 to 6

n = 6 =

n natural no = 1, 2, 3, 4, 5

n = 5

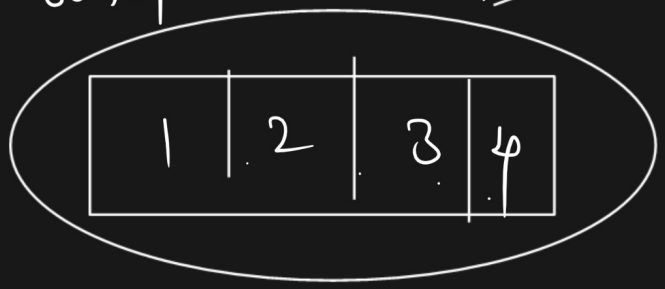
but one number betn 1 to n is missing

you have to find out that number?

Ans : = 5

missing = 1 ⇐ Ans

if array is sorted



given array is also not sorted format

```
for( int i = 1 ; i < n ; i++)
{
    if ( i != arr[i])
    {
        i = ?
    }
}
```

i = 1, 2, 3, 4, 5 !=