

add

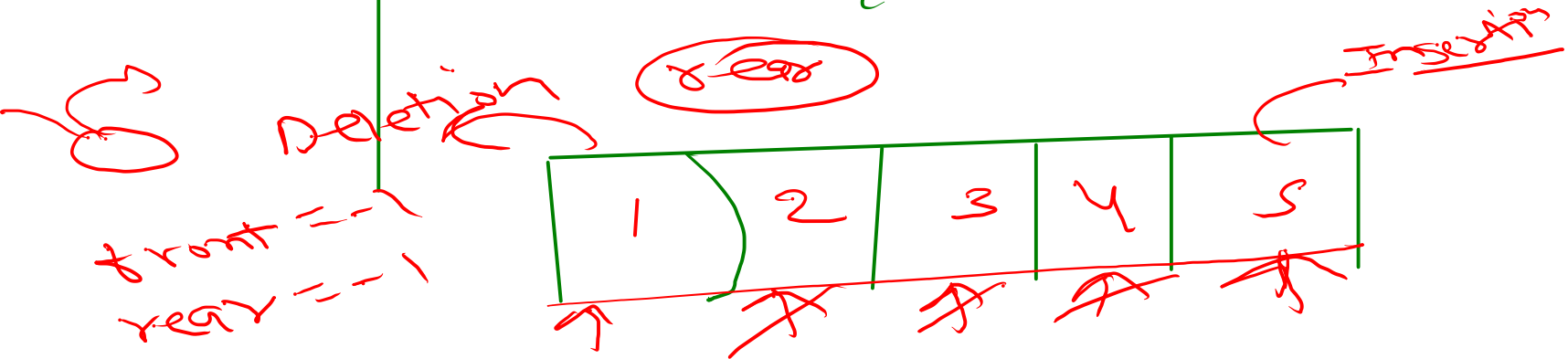
arr[rear+1] = x

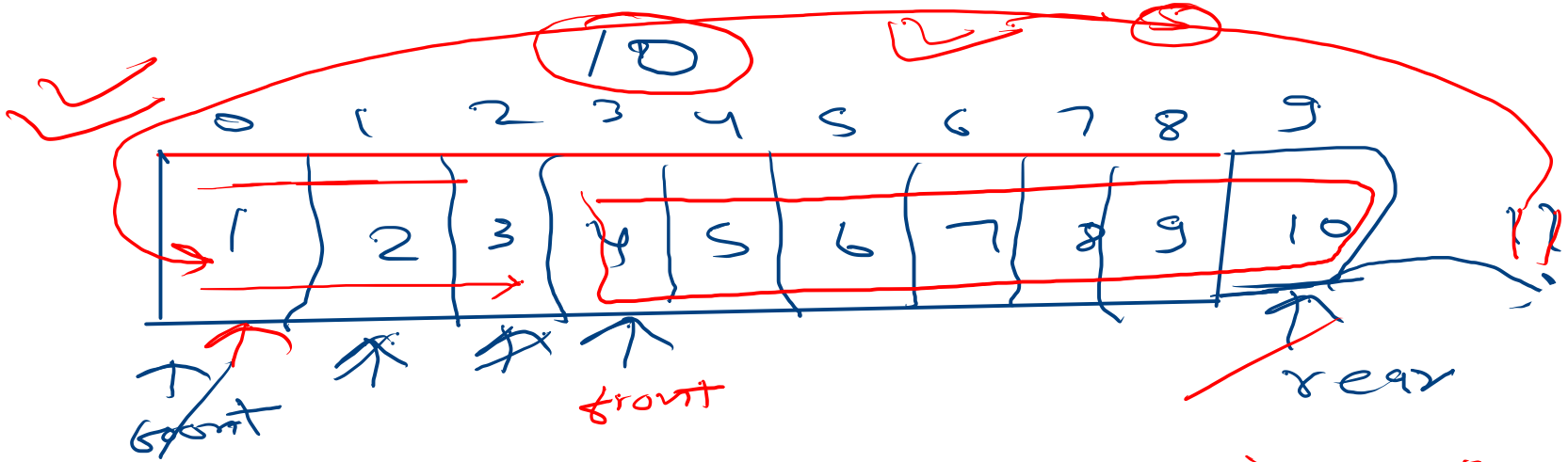
arr[0+1]

arr[1] = x

poll() / remove()

peek() / element()

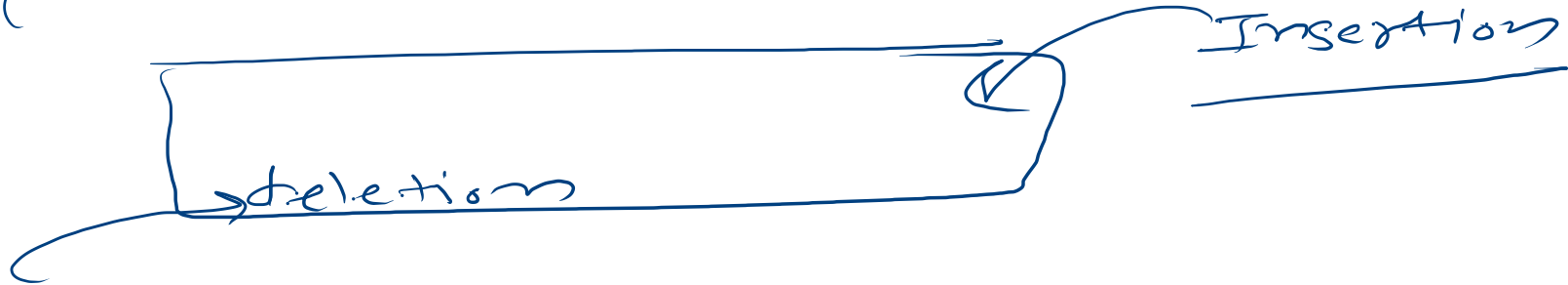
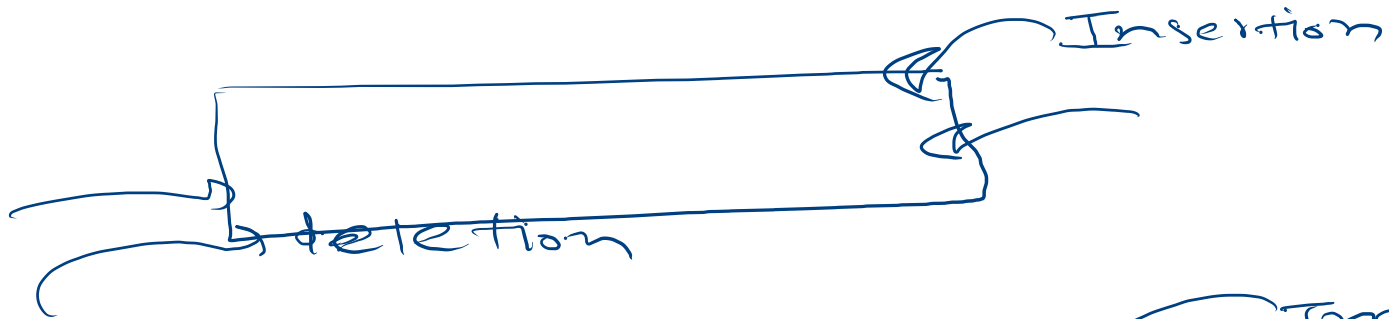




Circular Queue

size + 1  
size --

# Dequeue



+ add first LG

+ add last

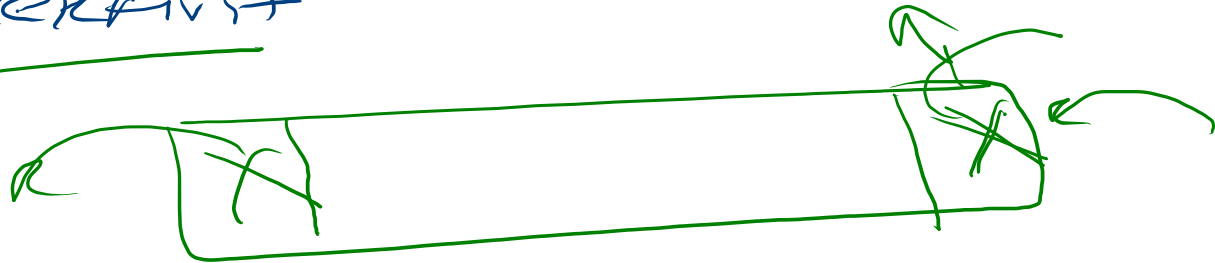
+ delete first

+ delete last

+ peek first

FIFO

FIFO



4

### Sample Output 0

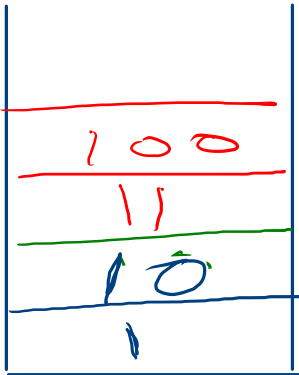
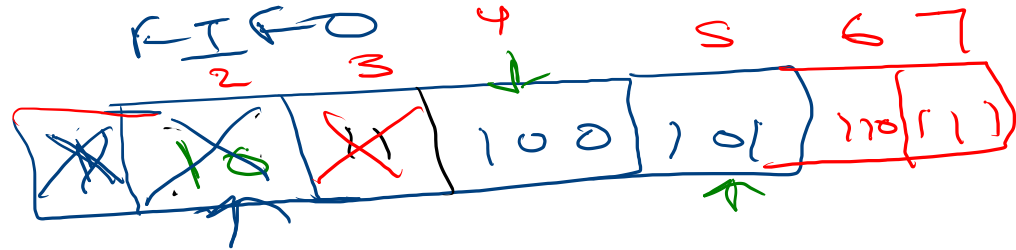
```
1 10 11 100
```

```

public static void binary(int n){
    ArrayList<String> arr = new ArrayList<>();
    Queue<String> queue = new LinkedList();
    queue.add("1");
    while(n-->0){
        String temp = queue.remove();
        arr.add(temp);
        queue.add(temp+0);
        queue.add(temp+1);
    }
    for(String x:arr){
        System.out.print(x+" ");
    }
}

```

4



Queue

temp = 11

11 11 + 0 → 110  
 11 11 + 1 → 111

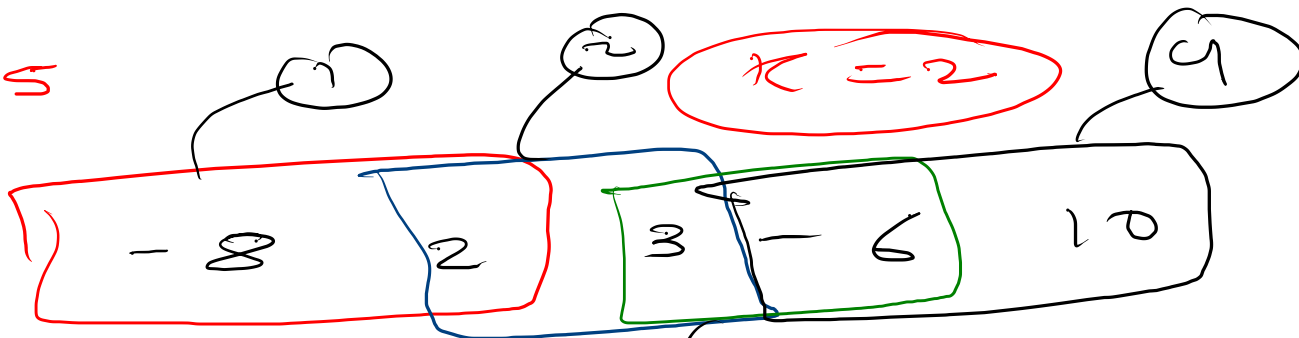


5 2

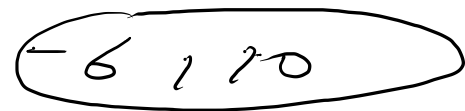
-8 2 3 -6 10



$N = 5$



3



-8 0 -6 -6



5 2  
-8 2 3 -6 10  
→

$$N = 5, K = 2$$

$\begin{matrix} 0 & 1 & 2 & 3 & 4 \\ -8 & 2 & 3 & -6 & 10 \end{matrix}$

$$N - K + 1$$

$$5 - 2 + 1$$

$$3 + 1 = 4$$

for  $(i = 0, i < N - K + 1)$

0 1 2 3

$[-8, 0, -6, -6]$

$$N = 5, k = 2$$

①  $N - k + 1$  — message

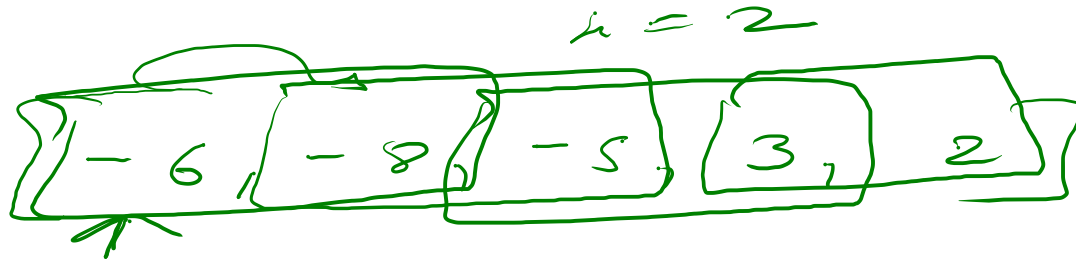
$$N - k + 1$$

$$5 - 2 + 1$$

②  $N - k + 1$

④

stopping condition



$[-6, -8, -5, 0]$

Input :

N = 8

A[] = {12, -1, -7, 8, -15, 30, 16, 28}

K = 3

Output :

-1 -1 -7 -15 -15 0

-1 -7, 8

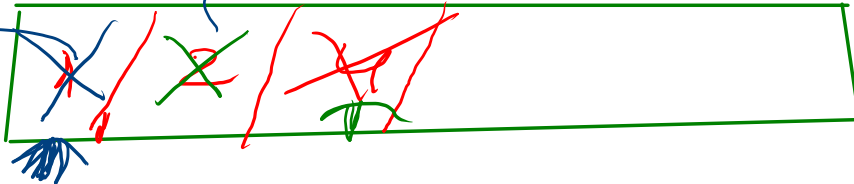
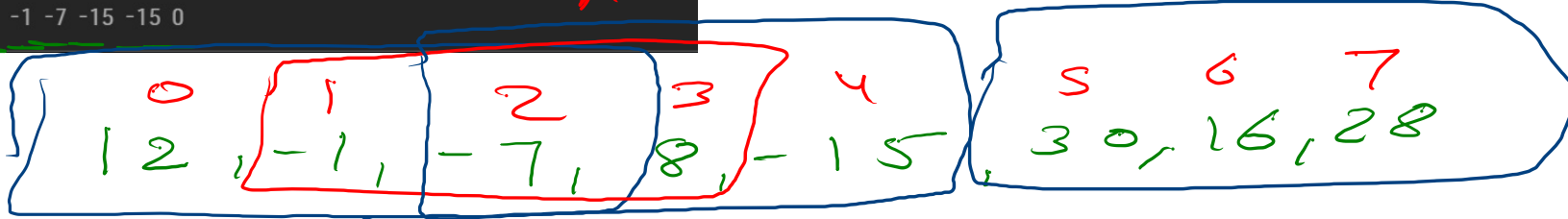
K = 3

-7, 8, -15

-15, 30, 16

8, -15, 30

30, 16, 28



-1 -1 -7 -15 -15 0