Update query 1

left = 2
right = 8
x = 0

Ex:-

n = 10

	0	1	2	3	4	5	6	7	8	9
arr =	1	2	3	4	5	6	7	8	9	10

2 8

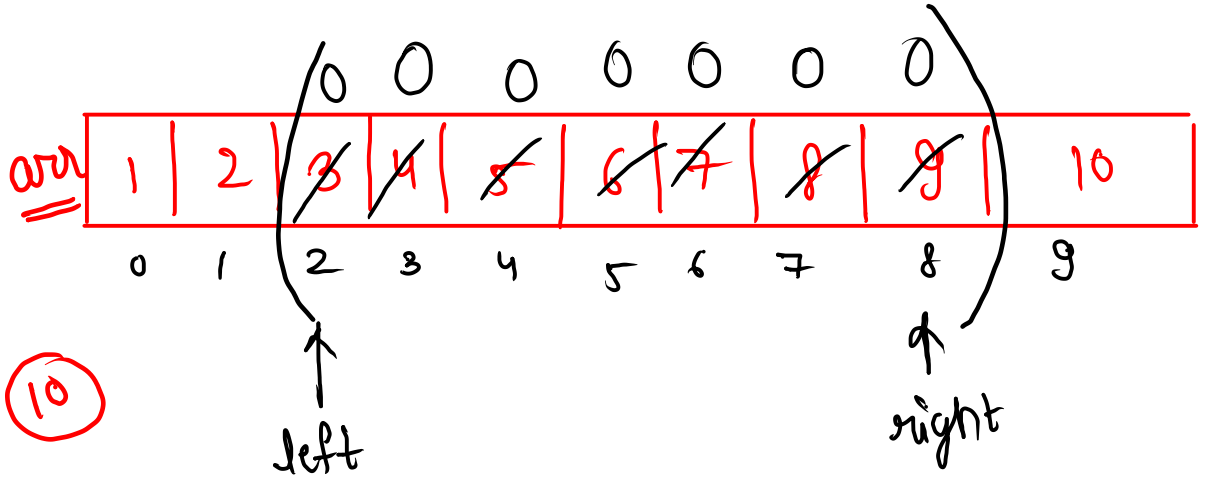
x = 0

$0 \leq 2, 8 \leq 9$

$-1 < 2, 8 < 10$

logic

update value of x from left to right both included



(10)

code

```java
public static void main(String[] args) {
    Scanner scn = new Scanner(System.in);
    int n = scn.nextInt();
    int[] arr = new int[n];
    for (int i = 0; i < n; i++) {
        arr[i] = scn.nextInt();
    }
    int left = scn.nextInt();
    int right = scn.nextInt();
    int x = scn.nextInt();

    int[] ans = updateArray(arr, left, right, x);
    for (int i = 0; i < ans.length; i++) {
        System.out.print(ans[i] + " ");
    }
}

public static int[] updateArray(int[] arr, int left, int right, int x) {
    for (int i = left; i <= right; i++) {
        arr[i] = x;
    }
    return arr;
}
```

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

ans | 1 | 2 | 0 | 0 | 0 | 7 |

arr

ans

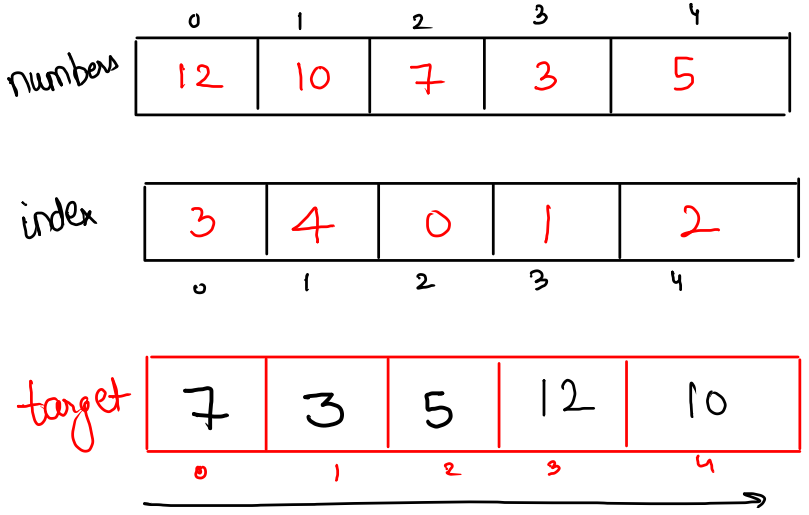
ans[i]

array return type

(n)

Solve Array

n = 5



i = 0, idx = 3, val = 12

i = 1, idx = 4, val = 10

i = 2, idx = 0, val = 7

i = 3, idx = 1, val = 3

i = 4, idx = 2, val = 5

logic

```
int idx = index[i];
int val = numbers[i];
target[idx] = val;
```

```java
public static void solveArray(int n, int[] number, int[] index ) {
    int[] target = new int[n];
    for (int i = 0; i < n; i++) {
        int idx = index[i];
        int val = number[i];
        target[idx] = val;

        // target[index[i]] = number[i];
    }

    for (int i = 0; i < n; i++) {
        System.out.print( target[i] + " ");
    }
}
```

1, 4, 3, 0, 2

100, 24, 37, 1, 101

val = number[3]

=

n = 5

numbers

0	1	2	3	4
100	24	37	1	101

index

0	1	2	3	4
1	4	3	0	2

i = 0,

i = 1,

i = 2,

i = 3,

i = 4

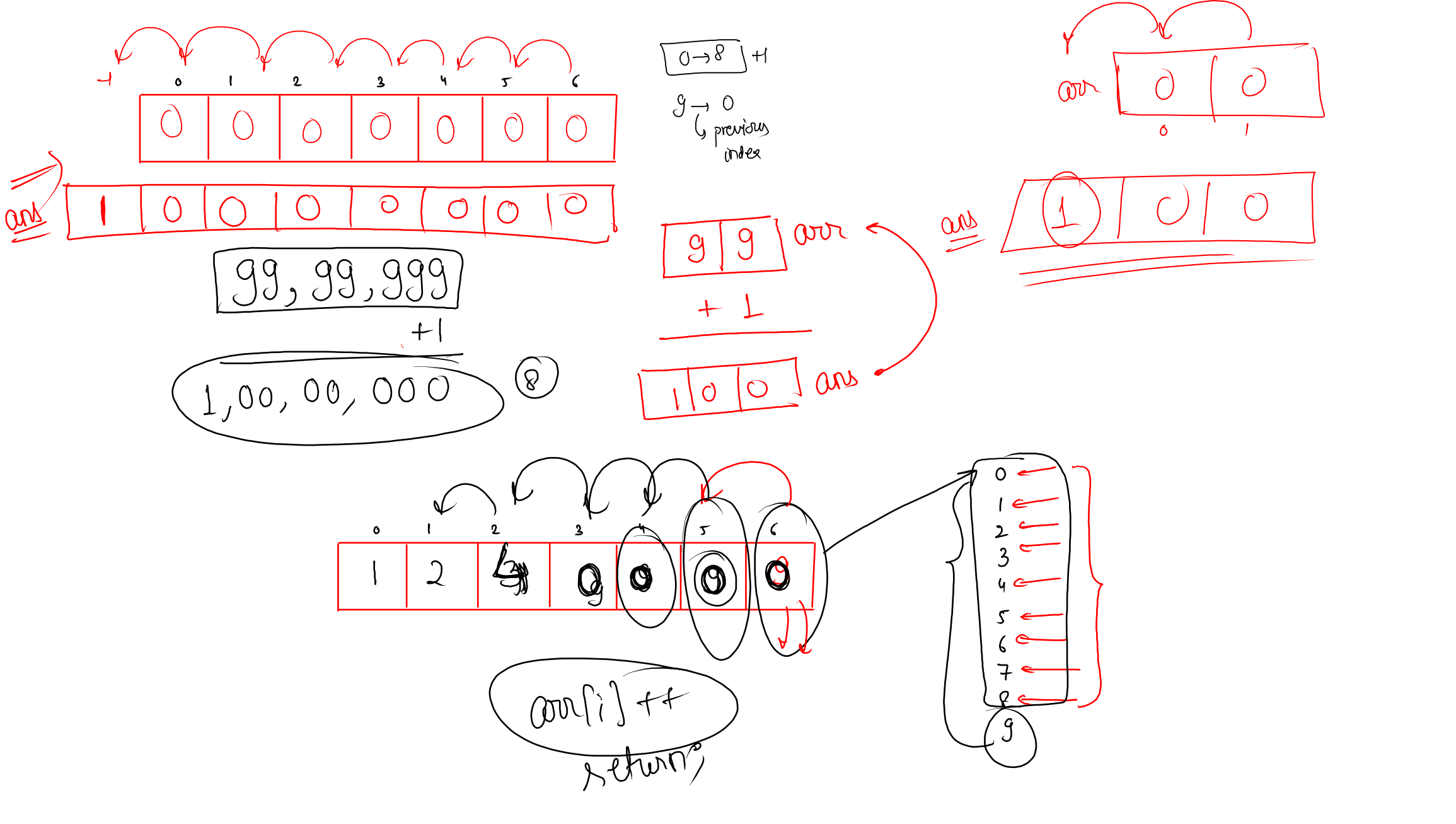
idx = 0

val = 1

target

0	1	2	3	4
1	100	101	37	24

target[0] = 1


0 1 2 3 4 5 6
0 0 0 0 0 0 0
ans
1 0 0 0 0 0 0 0
99,99,999
+1
1,00,00,000
8
0→8 +1
9 → 0
previous index
arr
0 0
0 1
9 9 arr
+ 1
1 0 0 ans
ans
1 0 0
0 1 2 3 4 5 6
1 2 9 0 0 0 0
arr[i]++
return;
0 1 2 3 4 5 6 7 8 9