GKSTR35 Count_Even

```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 ans = countEven(arr, n);
    System.out.println(ans);
}

public static int countEven(int[] arr, int n) {
    int count = 0;
    for (int i = 0; i < n; i++) {
        if ( arr[i] % 2 == 0 ) {
            count++;
        }
    }
    return count;
}
```
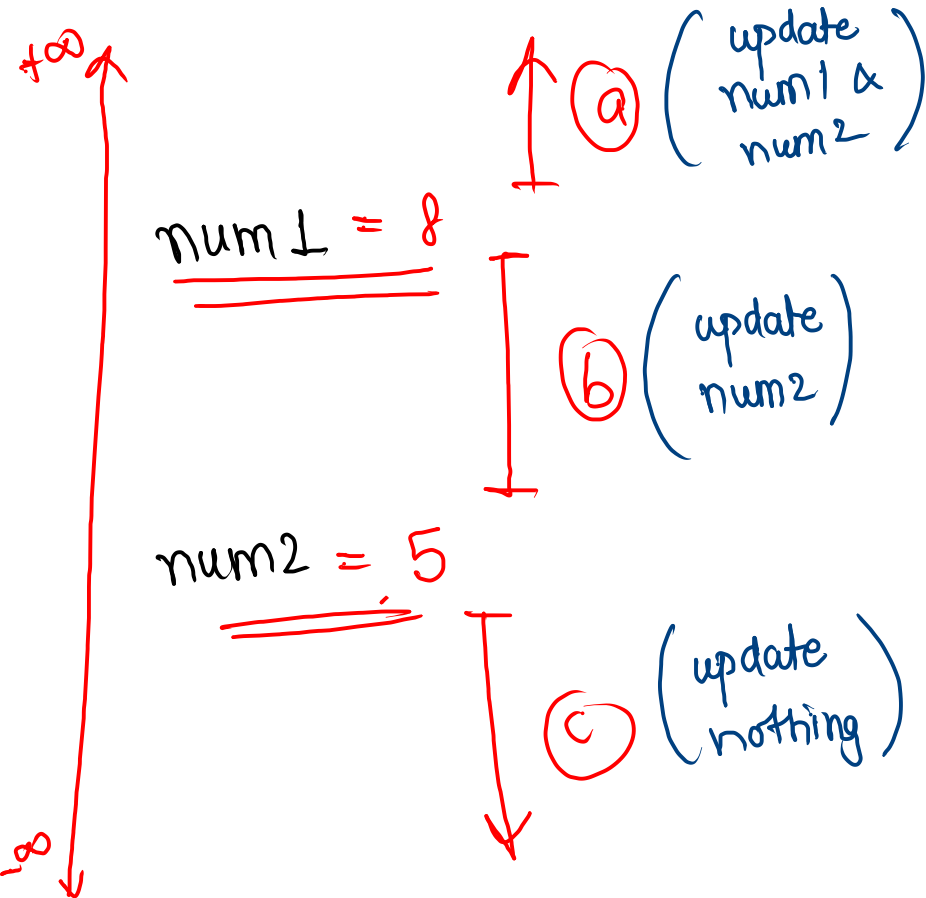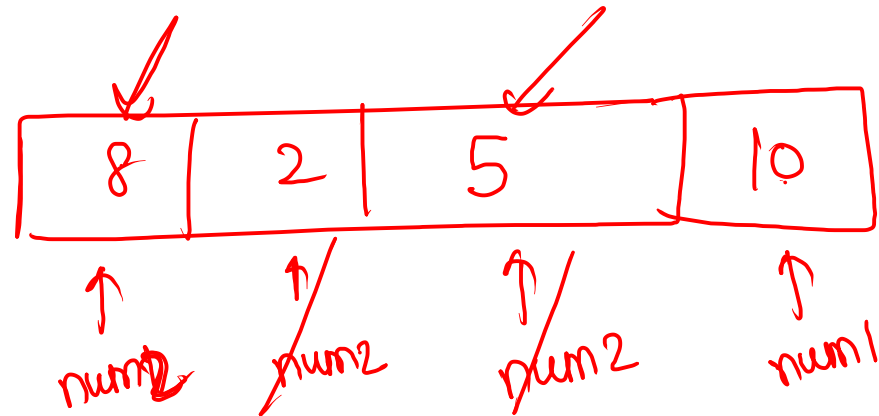
HW_Second Largest in array 2

arr	5	1	0	2	3	7	6	4	8	9	11	10	15	12	20

$+\infty$

ⓐ $\begin{pmatrix} \text{update} \\ \text{num1 \&} \\ \text{num2} \end{pmatrix}$

$num1 = 8$

ⓑ $\begin{pmatrix} \text{update} \\ \text{num2} \end{pmatrix}$

$num2 = 5$

ⓒ $\begin{pmatrix} \text{update} \\ \text{nothing} \end{pmatrix}$

$-\infty$

$arr[i] = 10$

8	2	5	10

num2   num2   num2   num1

arr	5	1	0	2	3	7	6	4	8	9	11	10	15	12	20

logic :-

$+\infty$

a

num1

b

num2

c

$-\infty$

area c : update nothing

area b : update num2 with current num (arr[i])

area a : update num2 with num1
        update num1 with current (arr[i])
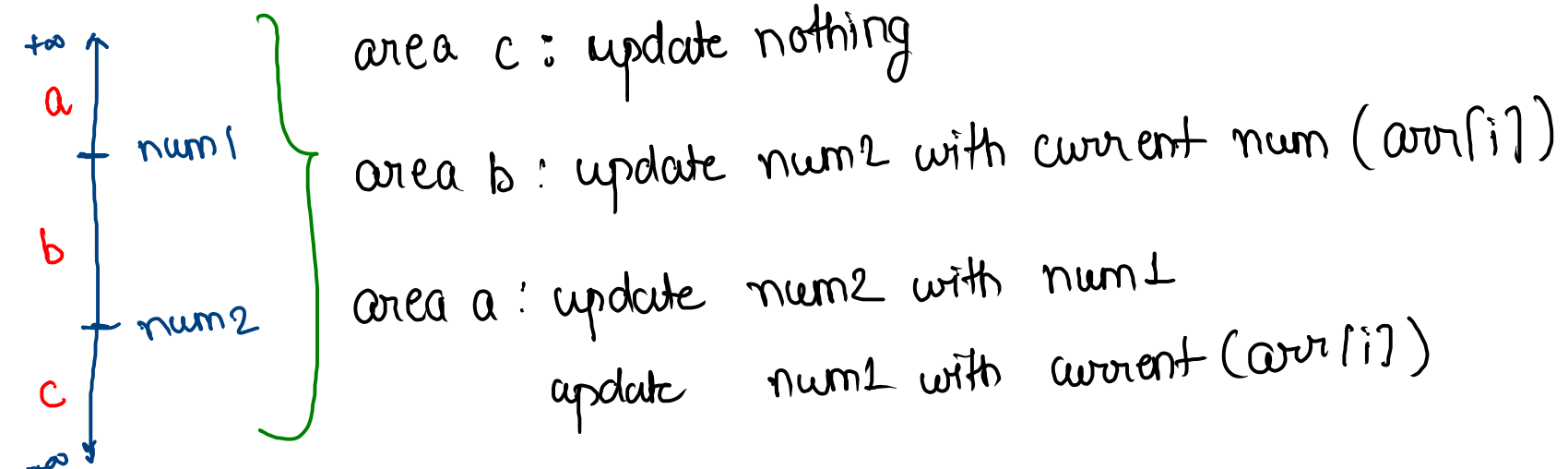
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14
arr	5	1	0	2	3	7	6	4	8	9	11	10	15	12	20

$arr[i] = $ ~~5~~ ~~1~~ ~~0~~ ~~2~~ ~~3~~ ~~7~~ ~~6~~ ~~4~~ ~~8~~ ~~9~~ ~~11~~ ~~10~~ ~~15~~ ~~12~~ 20

$num1 = $ ~~-∞~~ ~~5~~ ~~7~~ ~~8~~ ~~9~~ ~~11~~ ~~15~~ 20

$num2 = $ ~~-∞~~ ~~-∞~~ ~~1~~ ~~2~~ ~~3~~ ~~5~~ ~~6~~ ~~7~~ ~~8~~ ~~9~~ ~~10~~ ~~11~~ ~~12~~ 15

logic:-

area c : update nothing
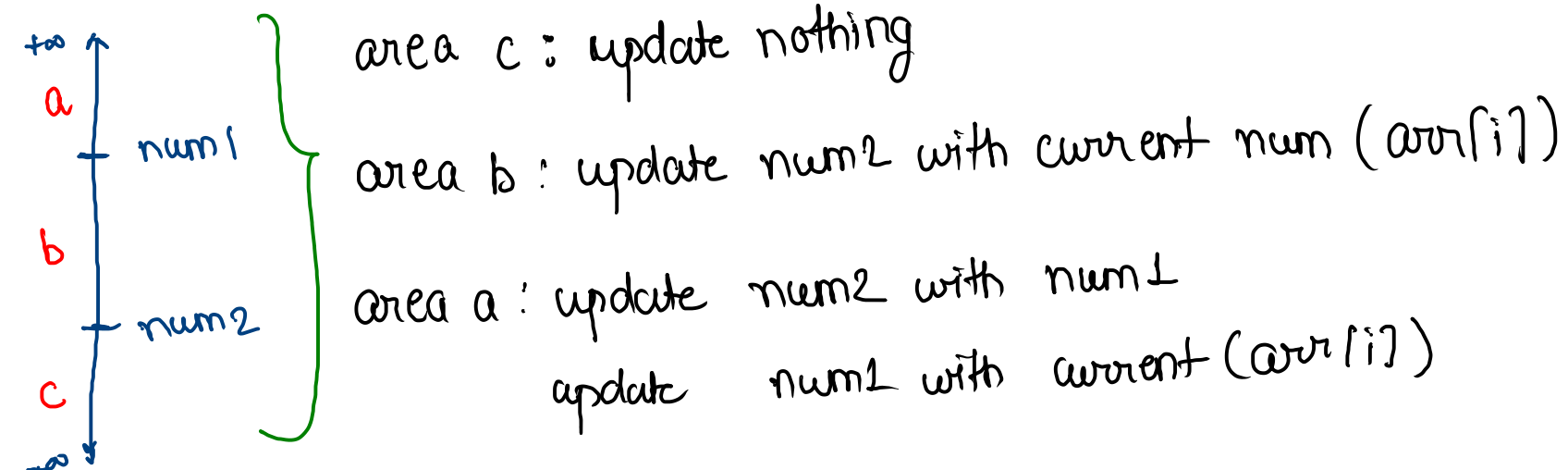
area b : update num2 with current num ($arr[i]$)

area a : update num2 with num1
         update num1 with current ($arr[i]$)

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();
    }


    secondLargest(arr, n);
}

public static void secondLargest(int[] arr, int n) {
    int num1 = Integer.MIN_VALUE; // largest number
    int num2 = Integer.MIN_VALUE; // second largest

    for (int i = 0; i < n; i++) {
        if ( arr[i] > num1 ) {   // a
            num2 = num1;
            num1 = arr[i];
        } else if ( arr[i] > num2 ) {   // b
            num2 = arr[i];
        }
    }

    System.out.println(num2);
}
```
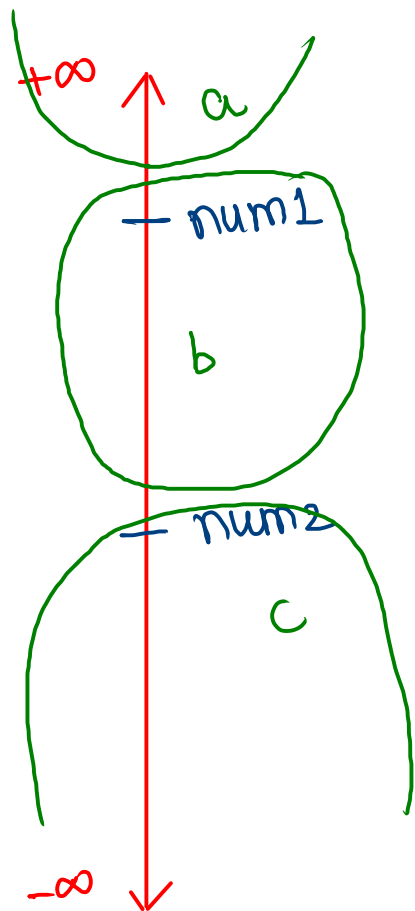
n = 4

90 8 90 5

num1 = $-\infty$ 90

num2 = $-\infty$ $-\infty$ 8

$+\infty$

a

num1

b

num2

c

$-\infty$

(arr[i] != num1)

```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();
    }
    secondLargest(arr, n);
}

public static void secondLargest(int[] arr, int n) {
    int num1 = Integer.MIN_VALUE; // largest number
    int num2 = Integer.MIN_VALUE; // second largest
    for (int i = 0; i < n; i++) {
a ->    if ( arr[i] > num1 ) {   // a
            num2 = num1;
            num1 = arr[i];
b ->    } else if ( (arr[i] > num2) && (arr[i] != num1) ) {   // b
            num2 = arr[i];
        }
    }
    System.out.println(num2);
}
```

Check Characterstic

```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();
    }

    for (int i = 0; i < n; i++) {
        if ( arr[i] > 0 ) {
            arr[i] = 1;
        } else if ( arr[i] < 0 ) {
            arr[i] = -1;
        } else if ( arr[i] == 0 ) {
            arr[i] = 0;
        }
    }

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