n = 4

→ 98464

arr = (int)

0	1	2	3
4	46	8	9

arr = (string)

0	1	2	3
"4"	"46"	"8"	"9"

a = "4"

b = "46"

a+b ("446") // lambda func.

b+a ("464")

arr = (string)

0	1	2	3
"9"	"8"	"46"	"4"

ans = " "

Convert int to string

1) string str .= String.valueOf(arr); ✓

2) String str = Integer.toString(arr); ✓

# a-b = } +ve → ↓ing
b-a = } -ve → ↑ing

str1 = - - .

str2 = - - -

```
return str1.compareTo(str2); // ↑ing
return str2.compareTo(str1); // ↓ing
```

arr =
(string)

0	1	2	3
"9"	"8"	"46"	"4"

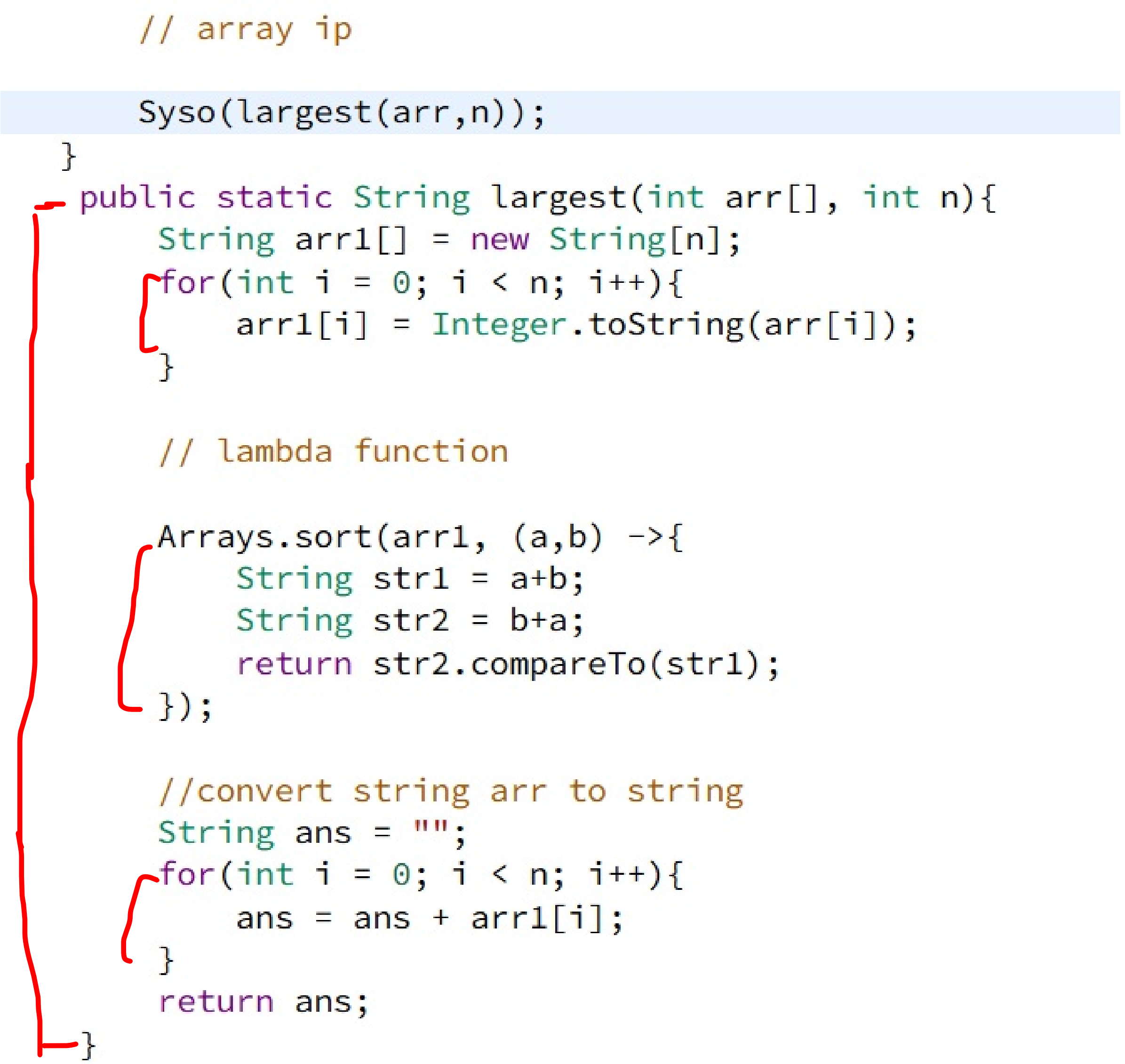
ans = " " + 9 = "9"

ans = "9" + "8" = "98"

"98" + "46" = "9846"

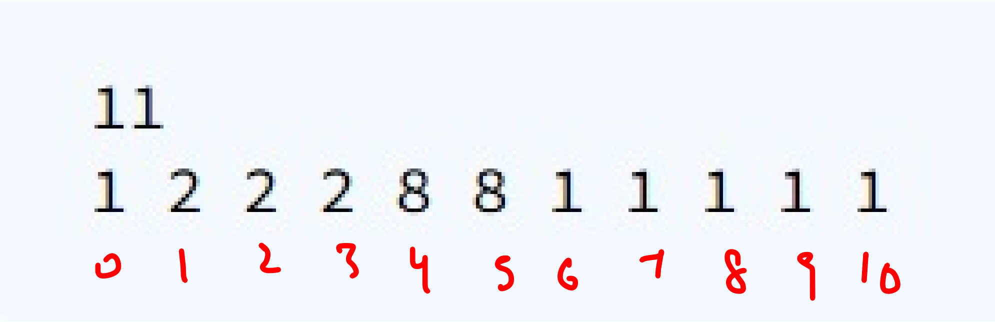
"9846" + "4"

=> "98464"



```
        // array ip

        Syso(largest(arr,n));
    }
    public static String largest(int arr[], int n){
        String arr1[] = new String[n];
        for(int i = 0; i < n; i++){
            arr1[i] = Integer.toString(arr[i]);
        }

        // lambda function

        Arrays.sort(arr1, (a,b) ->{
            String str1 = a+b;
            String str2 = b+a;
            return str2.compareTo(str1);
        });

        //convert string arr to string
        String ans = "";
        for(int i = 0; i < n; i++){
            ans = ans + arr1[i];
        }
        return ans;
    }
```

HW_Majority Element 10



```
11
1 2 2 2 8 8 1 1 1 1 1
0 1 2 3 4 5 6 7 8 9 10
```

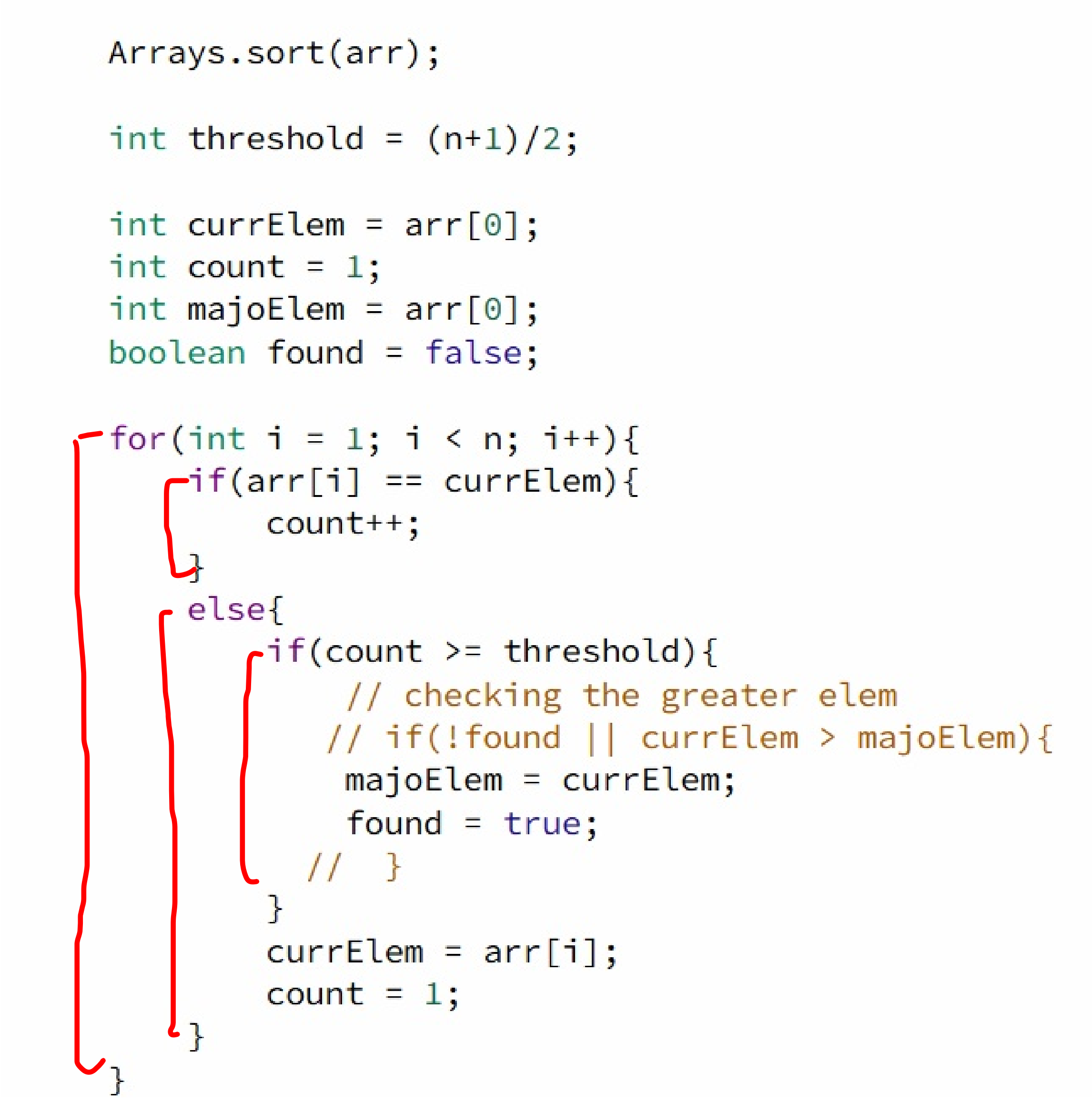
↓

1 | 1 | 1 | 1 | 1 | 1 | 2 | 2 | 2 | 8 | 8

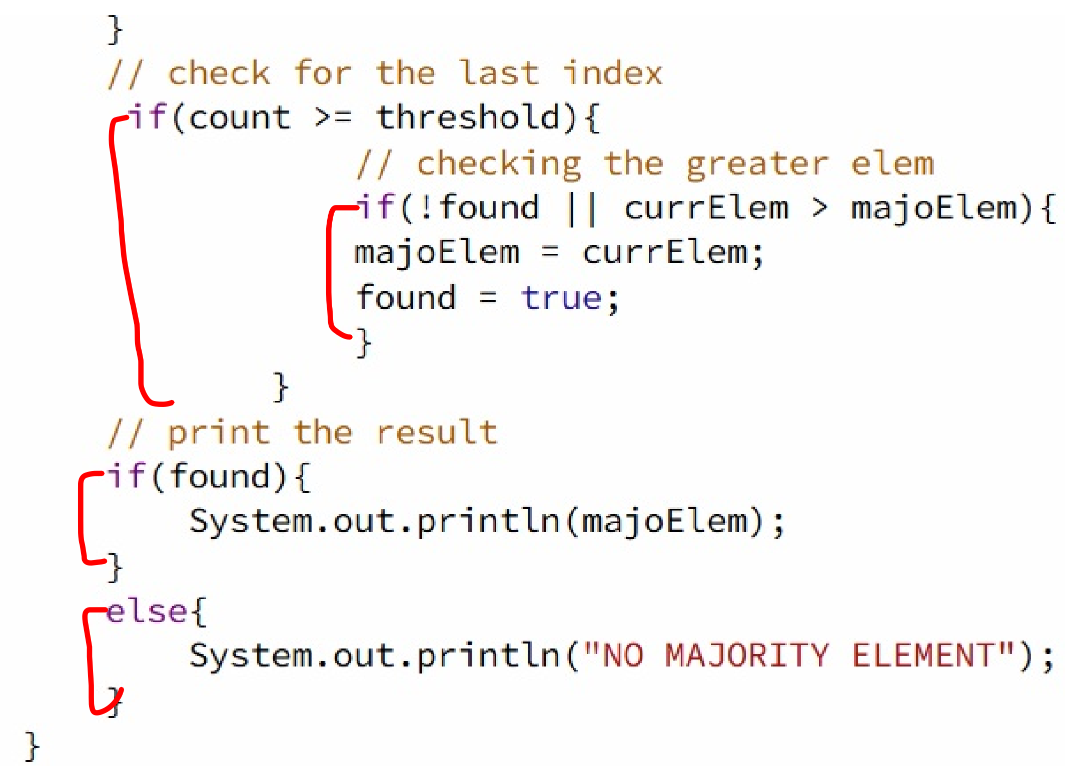
```
Curr = 1
count = 1
majelm = 1
found = f

(n+1)/2
```

```
i = 1
if arr[i] = curr
    1 == 1
    count++
else
if (count > (n+1)/2)
    maj = curr
    found = true
```

```
curr = arr[i]
count = 1;
```



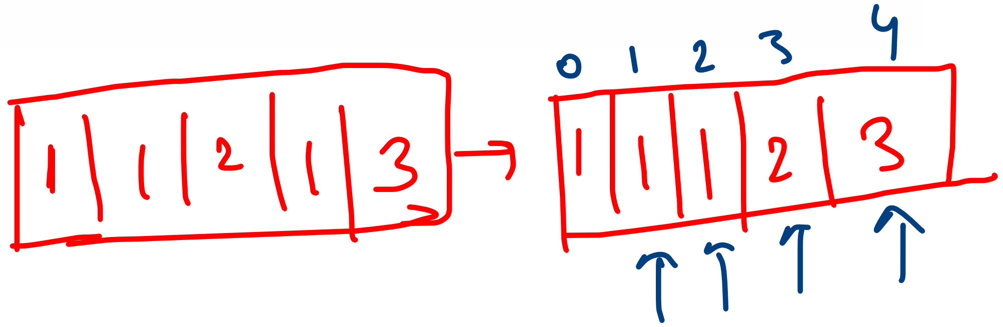
```
    Arrays.sort(arr);

    int threshold = (n+1)/2;

    int currElem = arr[0];
    int count = 1;
    int majoElem = arr[0];
    boolean found = false;

    for(int i = 1; i < n; i++){
        if(arr[i] == currElem){
            count++;
        }
        else{
            if(count >= threshold){
                // checking the greater elem
               // if(!found || currElem > majoElem){
                majoElem = currElem;
                found = true;
              //  }
            }
            currElem = arr[i];
            count = 1;
        }
    }
```



```
        }
        // check for the last index
        if(count >= threshold){
                    // checking the greater elem
                    if(!found || currElem > majoElem){
                    majoElem = currElem;
                    found = true;
                    }
            }
        // print the result
        if(found){
            System.out.println(majoElem);
        }
        else{
            System.out.println("NO MAJORITY ELEMENT");
        }
    }
```

n=5

arr =



curr = 1
count = 1
maj = 1
found = f

T 1 == 1
C = ~~2~~ 3 1 == 1
3 ≥ 3 T
maj = 1 ✓
f = t

2 == 1 F
curr = 2
c = 1

arr = 3 == 2
curr = 3
C = 1

O/P → 1