Ques Display Array

Sample Input

```
5
3
1
0
7
5
```

Sample Output

```
3
1
0
7
5
```

① Expectation: displayArr(int[] arr, int idx)

displayArr(arr, 0)

{3, 1, 0, 7, 5}

② Faith:

displayArr(arr, 1)

{1, 0, 7, 5}

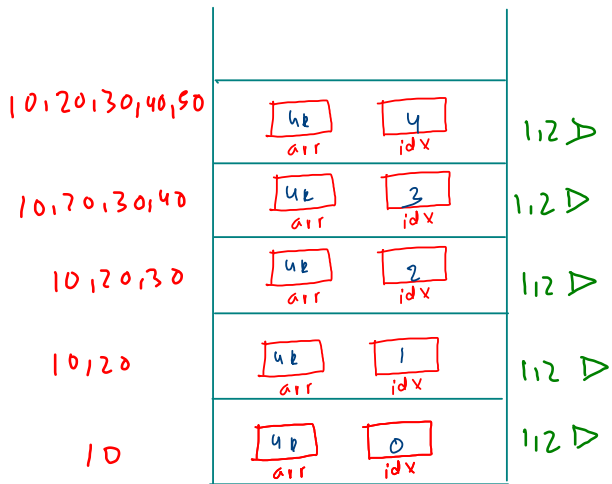
③ Meeting Expectation with faith

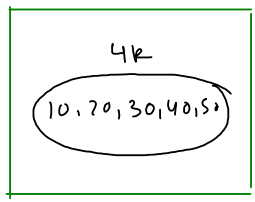
```
Syso(arr[idx]);
displayArr(arr, idx+1);
```

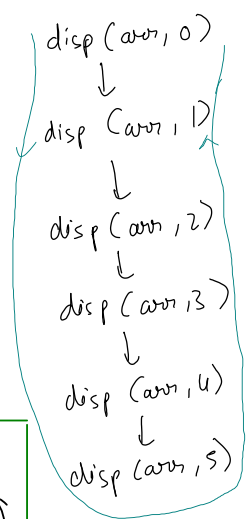
```
public static void displayArr(int[] arr, int idx){
    if(idx == arr.length) return;

    System.out.println(arr[idx]);
    displayArr(arr,idx+1);
}
```

0	1	2	3	4
10	20	30	40	50







10

10 20

10 20 30

10 20 30 40

10 20 30 40 50

Qw) Display Array in Reverse

{ 0:3, 1:1, 2:0, 3:7, 4:5 }

Sample Input

```
5
3
1
0
7
5
```

Sample Output

```
5
7
0
1
3
```

Expectation:

displayArrReverse (arr, 0);

```
5
7
0
3
1
```

faith:

displayArrReverse (arr, 1)

```
5
7
0
3
```

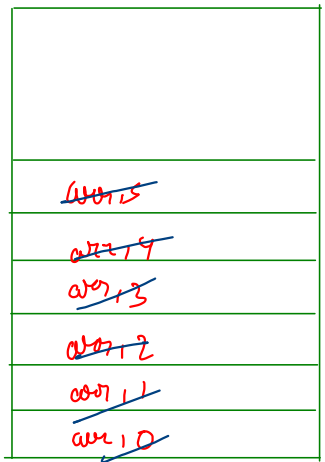
Meeting Expectation from faith

displayArrReverse (arr, idx+1);

Syso (n);

```
public static void displayArrReverse(int[] arr, int idx)    {
    if(idx == arr.length){
        return;
    }

    displayArrReverse(arr,idx+1);
    System.out.println(arr[idx]);
}
```



{ 0:1, 1:2, 2:3, 3:4, 4:5 }

1 ▷ 2 5

1 ▷ 2 5 4

1 ▷ 2 5 4 3

1 ▷ 2 5 4 3 2

1 ▷ 2 5 4 3 2 1

Ques) Maximum in An Array

{ 20, -30, 50, 10, 0, 60, 40 }

① Expectation:

int maxOfArray(int[] arr, int idx)

maxOfArray(arr, 0) → 60

② Faith:

maxOfArray(arr, idx+1)

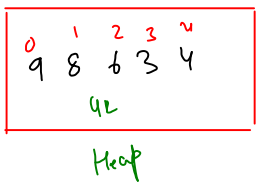
maxOfArray(arr, 1) → 50

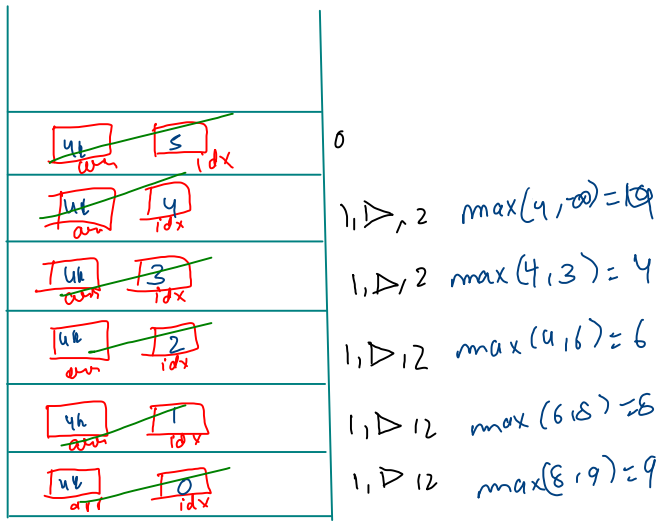
③ Meet Expectation with faith:

Compare our max with max that we get from the remaining array.

return max(arr[idx], maxCall)

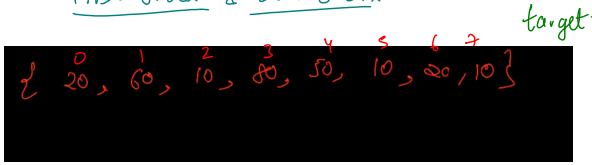
```
public static int maxOfArray(int[] arr, int idx){
  if(idx == arr.length) return Integer.MIN_VALUE;
  int maxOfRemArray = maxOfArray(arr,idx + 1);
  int totalMax = Math.max(arr[idx],maxOfRemArray);
  return totalMax;
}
```





First Index & Last Index

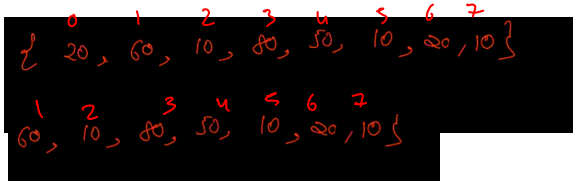
target = 10



First Index

① Expectation: fi(arr, 0) → 2

② faith: fi(arr, 1) → 2



③ Meet Expectation with faith

```
if (arr[idx] == target) return idx
else return fi(arr, idx+1);
```

```java
public static int firstIndex(int[] arr, int idx, int x){

    if(idx == arr.length) return -1;

    if(arr[idx] == x) {
        return idx;
    }

    return firstIndex(arr,idx + 1,x);
}
```

Last Index

① Expectation: li(arr, arr.length-1)

② faith: li(arr, idx-1)

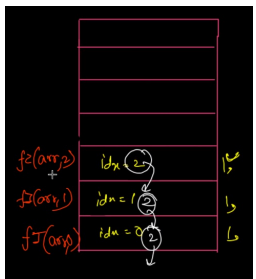
③ Meet Expe with faith:

```
if (arr[idx] == target)
    return idx;
else
    return li(arr, idx-1);
```

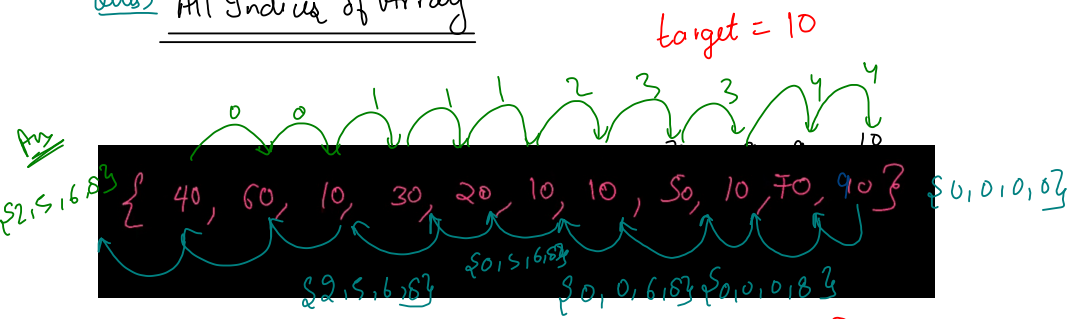
```java
public static int lastIndex(int[] arr, int idx, int x){
    if(idx == -1) return idx;

    if(arr[idx] == x) return idx;

    return lastIndex(arr,idx-1,x);
}
```



Ques) All Indices of Array



Expectation; allIndices (arr, target, idx, count)

10 0 0

{2,5,6,8}

faith: allIndices (arr, target, idx+1, count)

from idx+1 to arr.length

allIndices (arr, target, idx+1, count+1)

from idx+1 to arr.length + our own index element

Meeting Expectation with faith:

```java
public static int[] allIndices(int[] arr, int target, int idx, int count) {
    // write ur code here

    if(idx == arr.length) {
        int[] base = new int[count];
        return base;
    }

    if(arr[idx] == target) {
        int[] res = allIndices(arr,target,idx + 1,count + 1);
        res[count] = idx;
        return res;
    }

    int[] res = allIndices(arr,target,idx + 1,count);
    return res;
}
```

All Indices Dry Run

```
public static int[] allIndices(int[] arr, int target, int idx, int count) {
    // write ur code here

    if(idx == arr.length) {
        int[] base = new int[count];
        return base;
    }

    if(arr[idx] == target) {
        int[] res = allIndices(arr,target,idx + 1,count + 1);
        res[count] = idx;
        return res;
    }

    int[] res = allIndices(arr,target,idx + 1,count);
    return res;
}
```

0 1 2 3 4 5
{10, 20, 30, 10, 20, 10}

target = 10