

## Recursion in Arrays

Created By

Rah.



## Table of Contents.

- |    |                          |
|----|--------------------------|
| 1) | Display Array            |
| 2) | Display array in reverse |
| 3) | Map of an Array          |
| 4) | first Index              |
| 5) | Last Index               |
| 6) | All Indices of Array     |

```

public class displayarray {
    public static void disarr(int arr[], int size){
        if (arr.length == size){
            return;
        }
        System.out.println(arr[size]); ]→ 1
        disarr(arr, size+1); ]→ 2.
    }

    public static void main(String args[]){
        System.out.println("Enter the value of size of array");
        Scanner scn = new Scanner(System.in);
        int size = scn.nextInt();
        int arr[] = new int[size];
        for (int i = 0; i < size; i++){
            arr[i] = scn.nextInt();
        }
        disarr(arr, size, 0);
    }
}

```

Dry Run

array values [ 1, 2, 3, 4 ]

↳

{ arr[0] → 1 print .  
 arr[0+1] → print 2  
 arr[2] → print 3  
 arr[3] → print 4 .

## Display Array Contents -

### Logic

Step1 Take Input from the user

Step2 Pass the array and size to the function .

Step3 Recursively call the function till the time array size == function time

Step4 Print the value based on that .

Summary . when you need to print is imp  
 remember to understand pre & post order

```

package Recursion_array;

import java.util.Scanner;

public class reversedisplay {

    public static void revdis(int arr[], int val){
        if (val == 0)
            return;
        System.out.println(arr[val-1]);
        revdis(arr, val-1);
    }

    public static void main (String args[]){
        System.out.println("Enter the size of an array");
        Scanner scn = new Scanner(System.in);
        int size = scn.nextInt();
        int arr[] = new int[size];

        for (int i = 0 ; i < size ; i++){
            arr[i] = scn.nextInt();
        }
        revdis(arr,size);
    }
}

```

Dry Run .

→ Same logic as the previous question

$\text{arr} = 1, 2, 3, 4, 5$

→ ( $\text{arr}$ ,  $\text{arr.length}$ )

( $\text{arr}$ , 5)

↳ [ $\text{arr}[4]$ , 3] → Print (5)

[ $\text{arr}[3]$ , 2] → ↳

{ $\text{arr}[2]$ , 1} - 3

{ $\text{arr}[1]$ , 0} - 2

( $\text{arr}[0]$ , 0) -

## Reverse of An Array

Logic

→ Take the input from the user.

→ Now pass the size of an array and array to the function

→ Just like the previous one display the array from the last element to the first element

→ If previous question concepts are clear then this question is piece of cake .

Summary

```

package Recursion_array;

import java.util.Scanner;

public class maxofarray {

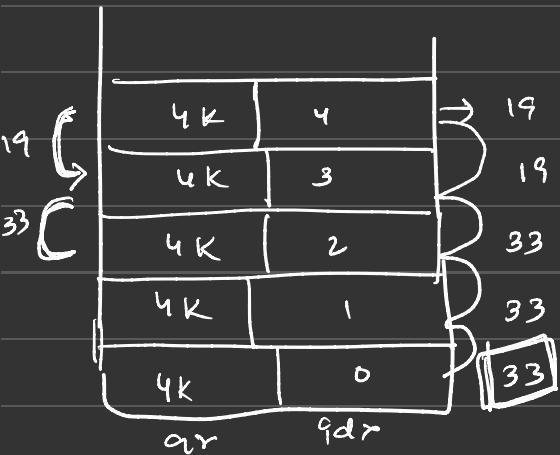
    public static int maxarr( int arr[] , int idx){
        if ( idx == arr.length -1){
            return arr[idx];
        }
        int maxa = maxarr(arr , idx+1);
        if ( arr[idx] > maxa){
            return arr[idx];
        }
        else {
            return maxa;
        }
    }

    public static void main (String args[]){
        System.out.println("Enter the size of an array");
        Scanner scon = new Scanner(System.in);
        int size = scon.nextInt();
        int arr[] = new int[size];
        for ( int i = 0 ; i < size ; i ++){
            arr[i] = scon.nextInt();
        }

        int val = maxarr(arr , idx: 0 );
        System.out.println("Max value is " + val);
    }
}

```

Dry Run .



## Max of an Array

find the maximum value  
of an Array .

Explained properly on the  
next page

$\text{arr} = [22, 2, 33, 7, 19]$

address  $\rightarrow$  4K , len = 5

$19 > 7 \rightarrow 19$

$19 < 33 \rightarrow 33$

$33 > 2 \rightarrow 33$

$33 > 22 \rightarrow 33$

## Summary

Learn How to establish  
proper faith & then what  
needs to be done .

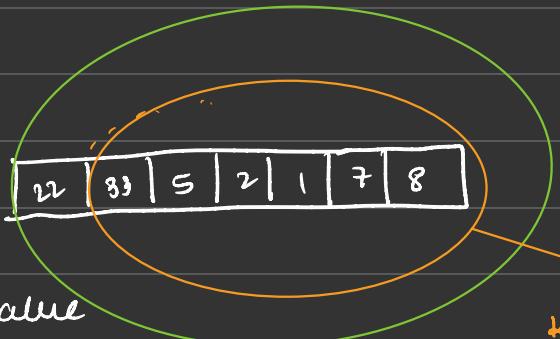
## Max of an Array

Expectation       $\text{Max}(\text{arr}, 0)$

→ Gives the maximum value from the array.

faith →  $\text{Max}(\text{arr}, 1)$

→ we know the maximum from  $(1 - \text{end})$



Now we just  
need to compare

it wothn the last value

so we  
know max  
of the array 1 - end  
is 33

and if the value is greater than  
the outcome pass that else  
keep the remaining value.

Max = Maxarr (arr, val+1)

if ( arr[val] > Max )  
    return arr[val+1]

else

Max

```

package Recursion_array;
import java.util.Scanner;
public class firstIndex {
    public static int fstdx(int arr[], int idx, int val) {
        if (idx == arr.length - 1) {
            return -1;
        }
        if (arr[idx] == val) {
            return idx;
        }
        else {
            int max = fstdx(arr, idx + 1, val);
            return max;
        }
    }

    public static void main (String args[]){
        System.out.println("Enter the size of an array");
        Scanner scn = new Scanner(System.in);
        int size = scn.nextInt();
        int arr[] = new int[size];
        for (int i = 0 ; i < size ; i ++){
            arr[i] = scn.nextInt();
        }
        System.out.println("Enter the value to find");
        int val2 = scn.nextInt();

        int val = fstdx(arr, 0 , val2);
        System.out.println("Index value is " + val);
    }
}

```

## first Index & last Index

Given a array find the values  
first Index.

Step 1

Take the array Input  
from the user

Step 2 Take the value input  
to find in the array

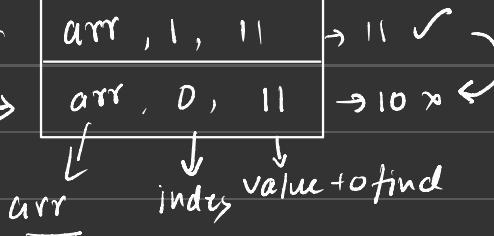
Step 3 call the recursive  
function from 0 - end and  
return the value if found  
very first time

Summary

Not that tough Question

10, 11, 11, 12, 13, 14

Return !



## first Index of an array

arr = 10, 11, 11, 12, 11, 13, 14

Expectation → `fstidx(arr, qdr, cnt)`

this will provide the value from the first idr

Value to find

qdr = 11

10, 11, 11, 11, 12, 11, 13, 14

- 1) Checking all the values from 0 index to end.
- 2) If found the value return the value else -1

for last index in the function from the back to the front.

```

public static int[] allind( int arr[], int idx , int val , int fsf){
    if ( idx == arr.length){
        return new int[fsf];
    }
    if (arr[idx] == val){
        int[] iarr = allind(arr , idx+1 , val , fsf+1);
        iarr[fsf] = idx;
        return iarr;
    }
    else{
        int[] iarr = allind(arr , idx+1 , val , fsf );
        return iarr;
    }
}

```

]-1

]-2

]-3

## All Indices of an Array .

find all the indices of an array and return the array with each location .

### Step1

Traverse through whole array and if found the value then  
 increase iarr index and  
 If didnt find the value  
 then continue traversing  
 the list

Dry Run .

### Summary

Need NO explanation its  
 plain & simple .

