



"समय न लगाएँ इसमें कि क्या करना है,  
वरना समय ये तय करेगा कि आपका क्या करना है।"

# ARRAY DATA STRUCTURE

CONCEPTS

REQUIREMENT

PROBLEMS

APPROACH

## Welcome & Vision

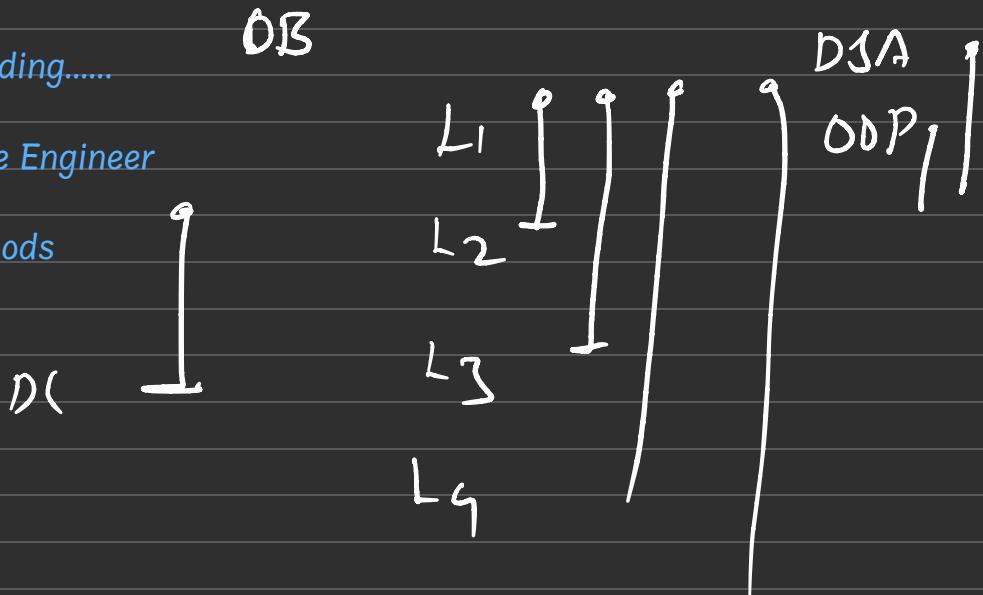
30 40 400 3hr 15

Why you started this course. — 2 WDSA → 1 DOPs → 2 WDSA

"I don't just want you to learn coding.....

Don't Mug up - You are Engineer

Change Studying methods



## Course Roadmap

- ✓ Roadmap
  - ✓ HomeWork Problems
  - ✓ Class Code
  - ✓ Notes
  - ✓ Weekly Tests
  - Mega Tests (5)  $\Rightarrow$
  - Mock Interview (3)  $\Rightarrow$  2 individuals | Group
- pdf  
code

CPP - STL

Java - Collection

sort( $a^n$ )

Pre-requisite - STL or Collections

✓ How to Approach the problem

✓ Code Quality

int & int arr

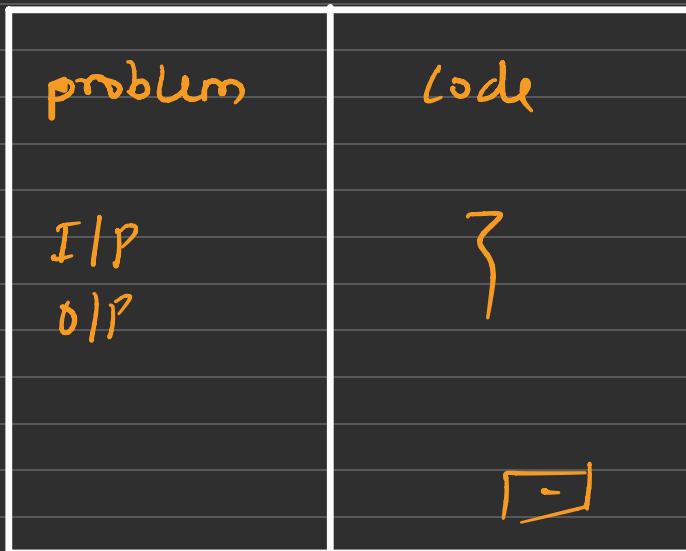
Total Time = 45 min

5 min = Read

3/5 min = Sample I/P

30 min = Logic

- X —
- Hint | Topic (10 min)
  - Solution
  - go & Read other



void sumOfTwoNo()

}

20,000

Resign

Diksha

void fun()

}

10,000 line

Siddhi

↳

# Sum of two Number

sumOfTwoNumber → camel case

int arr ()      int num ()

maximum

# Basic Math

✓ Divisors

✓ Prime Numbers

Divisor:

$$\begin{array}{r} 10 : \quad | \quad 2 \quad 5 \quad 10 \quad \underline{\text{Divisors}} \\ 1 \overline{)10} \qquad 2 \overline{)10} \qquad 5 \overline{)10} \qquad 10 \overline{)10} \\ \underline{0} \qquad \underline{0} \qquad \underline{0} \qquad \underline{0} \end{array}$$

a No. which completely divides given number,  
is Divisor

$$10 \% 3 = 1$$

$\%$  = Remainder

$$9 \% 3 = 0$$

$N \rightarrow 1 \text{ to } N$

$$N=8$$

1 2 4 8 --

$N=100$

4  
4%1 =  
4%2 =  
4%3 =  
4%4 =  
7

```
for (i=1 ; i < N ; i++)  
    if (N % i == 0)  
        printf("%d")
```

Y

N

Y

## Prime Number:

Number which has only 2 divisor  
that is 1 & no. itself

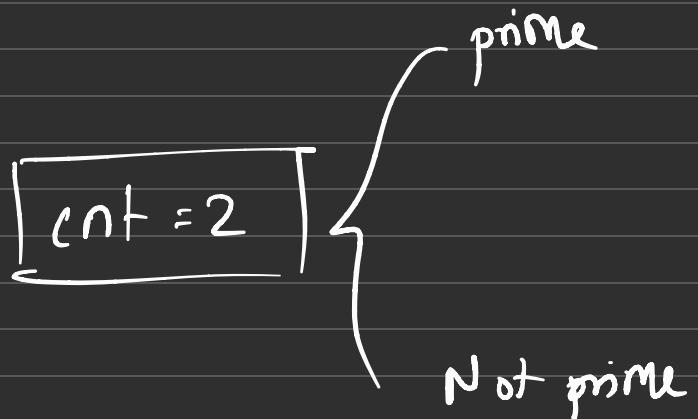
$$(2) = 1, 2$$

$$(3) = 1, 3$$

$$(5) = 1, 5$$

$$(7) = 1, 7$$

$$(11) = (1, 11)$$



# Arrays

| |

100

Why we need Array

int  $x_1 = 10$

int  $x_2 = 30$



Siddhi Diksha

$x_1$	$x_2$	$x_3$
$x_4$	$x_5$	$x_6$
$x_7$	$x_8$	$x_9$

— — —  $x_{100}$

int arr = {10, 20, 30}

arr = {  $x_1, x_2, x_3$  } (B)

$x_{40} = Me$

$x_{55} = Prachi$

## What is Array

*Collection of similar elements stored in continuous memory allocation.*

int : 4B

int arr(N)

arr(3)

\* (arr + 3)

\* (100 + 3)

\* (100 + 3 \* 4)

\* (100 + 12)

\* (112) = 4

arr

100	104	108	112	116
1	6	2	4	3
0	1	2	3	4

$$arr[2] = 2$$

$$arr[3] = 4$$

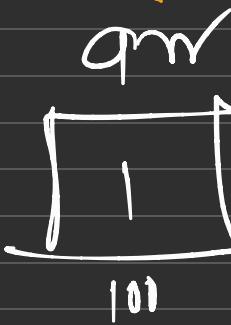
~~void fun( int arr() )~~



void fun( int \*arr )

int arr()

fun(arr)



*What if I tell you.. There is no concept called as Array*

## Largest Element in Array



Difficulty: Basic

Accuracy: 67.48%

Submissions: 510K+

Points: 1

Average Time: 20m

Given an array **arr[]**. The task is to find the **largest** element and return it.

### Examples:

**Input:** arr[] = [1, 8, 7, 56, 90]

= | 7    8    56    90

**Output:** 90

**Explanation:** The largest element of the given array is 90.

**Input:** arr[] = [5, 5, 5, 5]

**Output:** 5

**Explanation:** The largest element of the given array is 5.

**Input:** arr[] = [10]

**Output:** 10

**Explanation:** There is only one element which is the largest.

- Empty
- sorted | unsorted
- duplicates

	7	6	10	19	12
	0	1	2	3	4

① sort array & return last element

{	6	}	7	}	10	}	12	}	19	}
	0		1		2		3		4	

$O(n-1)$

7	6	10	19	12
0	1	2	3	4



$$\max = \begin{cases} \text{MIN} \\ \text{MAX} \end{cases}$$

~~max = INT. MIN~~

~~7 10~~

~~19~~

ans

find max = MIN

find min = MAX

int largest ( arr[], N)

}

max = INT\_MIN

for( i=0 : i < n : i++)

    3

        if (a(i) > max)

            3 max = a(i)

        4

    5

        return max

5

## Second Largest



Difficulty: **Easy**

Accuracy: **26.72%**

Submissions: **1.3M**

Points: **2**

Average Time: **15m**

Given an array of **positive** integers **arr[]**, return the **second largest** element from the array. If the second largest element doesn't exist then return **-1**.

Note: The second largest element should not be equal to the largest element.

FL

### Examples:

**Input:** arr[] = [12, 35, 1, 10, 34, 1]

**Output:** 34

**Explanation:** The largest element of the array is 35 and the second largest element is 34.

5 5 5 5

**Input:** arr[] = [10, 5, 10]

**Output:** 5

**Explanation:** The largest element of the array is 10 and the second largest element is 5.

7	6	10	1	19	12
---	---	----	---	----	----

0 1 2 3 4

(N-1)

sort :

6	7	10	12	19
---	---	----	----	----

0 1 2 3 4

7	6	10	19	12	19	19
0	1	2	3	4		

Sort:

$(N-1)$

6	7	10	12	19	19	19
0	1	2	3	4	5	6

```
int secondLargest (arr, n)
```

```
}
```

```
sort (arr)
```

6	7	10	12	19	19	19		

0 1 2 3 4 5 6

```
for (n-2; i >= 0; i--)
```

```
if (arr[i] = arr[i+1])
```

```
return arr[i]
```

5 5 5  
0 1 2

i = 0 ->

Return -1

```
}
```

6	7	10	12	19	19	19
0	1	2	3	4	5	6

↓

$\Rightarrow 12$

first Largest = largest (arr)

second Largest = -1

5	5	5	5
0	1	2	3

↓

$\Rightarrow -1$

$$fL = 19$$

$$SL = 16 \neq 12$$



$$FL = 5$$

$$CL = -1$$

```
int secondLarge(arr, n)
```

```
}
```

```
int firstLargest = largest(arr)
```

```
int secondLargest = -1
```

```
for ( i=0 ; i<n ; i++ )
```

```
    if ( arr(i) > secondLargest ) { }
```

```
        arr(i) = firstLargest )
```

```
}
```

```
secondLargest = arr(i)
```

```
    }
```

```
    }
```

```
return secondLargest
```

## Rotate Array to right by One Position

1	2	3	4	5	6
0	1	2	3	4	5

→ Rotate by 1 position  
· To Right

6	1	2	3	4	5
0	1	2	3	4	5

	1	2	3	4	5	6
0	1	2	3	4	4	5



$$i = \emptyset$$

$$\times a(\underline{i+1}) = \emptyset$$

$$\times a(i) = a(i+1)$$

$$\times a(i) = a(i-1)$$

~~X~~

6	6	2	3	4	5
1	2	3	4	5	6

0 1 2 3 4 5

6	1	2	3	4	5
1	2	3	4	5	6

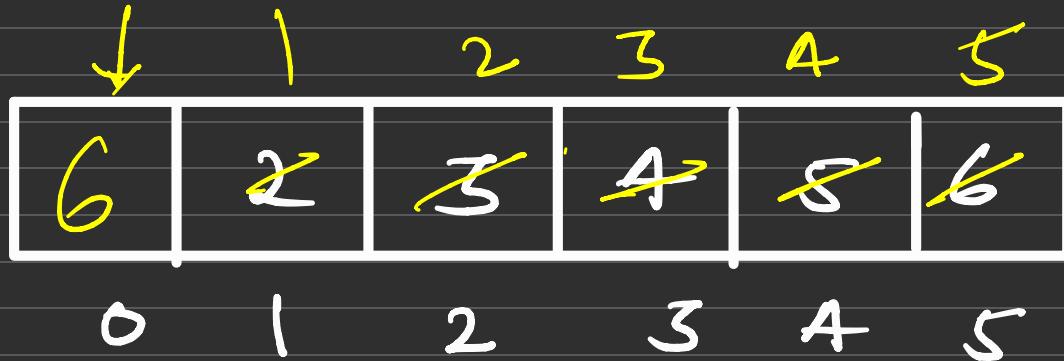
0 1 2 3 4 5

6	1	2	3	4	5
2	1	3	4	5	6

0 1 2 3 4 5

temp = 1

$$\text{temp} = q(n-1)$$



$$q(i+1) = q(i)$$

```
void rotate (arr)
{
    temp= arr[n-1]
    for ( i=n-2 ; i>=0 ; i--)
    {
        arr[i+1] = arr[i]
    }
}
```

$$arr[0] = temp$$

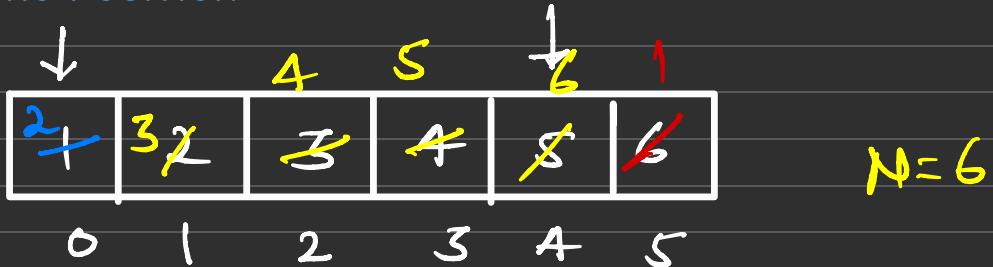
6	1	2	3	4	5
1	2	3	4	5	6

0 1 2 3 4 5

$$temp = 6$$

$$i = 4 \neq 2 + 0 - 1$$

## Rotate Array to left by One Position



$$i=0 \Rightarrow a(0) = a(1)$$

~~temp = arr[0]~~

$$i=1 \Rightarrow a(1) = a(2)$$

$a(i) = a(i+1)$

$$i=2 \Rightarrow a(2) = a(3)$$

$$i=3 \Rightarrow a(3) = a(4)$$

1 2 3 4 < 5

$$i=4 \Rightarrow a(4) = a(5)$$

```
void leftRotate (arr)
{
    temp = arr[0]
```

```
for ( i=0 ; i < n-1 ; i++ )
```

```

}
    arr[i] = arr [i+1]
```

```
}
```

```
arr[n-1] = temp
```

```
}
```

$K = \text{No. of Rotation}$

1	2	3	4	5	6
0	1	2	3	4	5

$K=1$

2 3 4 5 6 1

$K=2$

3 4 5 6 1 2

$K=3$

4 5 6 1 2 3

$K=4$

5 6 1 2 3 4

$K=2$  3 4 5 6 1 2 Ans

while ( $K > 0$ )

}

$\text{temp} = \text{arr}(0)$

. for ( $i=0$  ;  $i < n-1$  ;  $i++$ )

}

$\text{arr}(i) = \text{arr}(i+1)$

{

$\text{arr}(n-1) = \text{temp}$

$K--$

}

# 189. Rotate Array

Solved 

Medium

Topics

Companies

Hint

Re-do

Given an integer array `nums`, rotate the array to the right by `k` steps,  
where `k` is non-negative.

## Example 1:

Input: `nums = [1,2,3,4,5,6,7]`, `k = 3`

Output: `[5,6,7,1,2,3,4]`

### Explanation:

rotate 1 steps to the right: `[7,1,2,3,4,5,6]`

rotate 2 steps to the right: `[6,7,1,2,3,4,5]`

rotate 3 steps to the right: `[5,6,7,1,2,3,4]`

1	2	3	4	5	6
0	1	2	3	4	5

k=3

k=1

6 1 2 3 4 5

k=2

5 6 1 2 3 4

k=3

4 5 6 1 2 3

6	1	2	3	4	5
✓	✗	✗	✗	✗	✗

$N = 6$

0 1 2 3 4 5

$K=1$

$\text{temp} = arr(n-1)$

for ( $i=n-2; i \geq 0; i--$ )

?

$arr(i+1) = arr(i)$

$\text{temp} = arr(N-1)$

$arr(i+1) = arr(i)$

?

$i = 4 \quad arr(5) = arr(4)$

$arr(0) = \text{temp}$

$i = 3 \quad arr(4) = arr(3)$

$i = 2 \quad arr(3) = arr(2)$

$i = 1 \quad arr(2) = arr(1)$

$i = 0 \quad arr(1) = arr(0)$

while ( $k > 0$ )

}

    temp = arr[n-1]  
    for (i=n-2; i ≥ 0; i--)

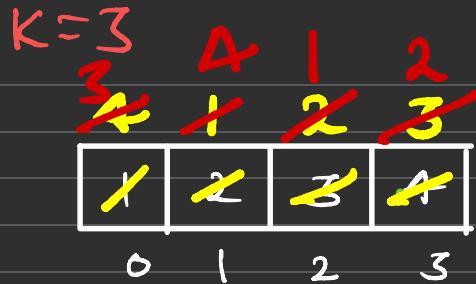
        ?     arr[i+1] = arr[i]

        ↳

        arr[0] = temp

        K --

        ↳     K = 1



K = 3

i = 2     arr[3] = arr[2]

i = 1     arr[2] = arr[1]

i = 0     arr[1] = arr[0]

K = 2

temp = 3

i = 2     arr[3] = arr[2]

i = 1     arr[2] = arr[1]

i = 0     arr[1] = arr[0]

$K = K \% N$

$K \% N$

1	2	3	4	5
0	1	2	3	4

$N = 5$



$$N = \{0, 1, 2, \dots, N-1\}$$

$$K = 20$$

$N=5$



Reverse = 5 4 3 2 |  
 $(0 - k-1)$

Reverse first = 3 4 5 2 |

$k$  Element

$(k - N-1)$

Reverse Rcm. 3 4 5 → 2

Elcm

K = 10

K % N = 0

## 283. Move Zeroes

Solved 

Easy

Topics

Companies

Hint

Re-do

Given an integer array `nums`, move all `0`'s to the end of it while maintaining the relative order of the non-zero elements.

**Note** that you must do this in-place without making a copy of the array.

### Example 1:

Input: `nums = [0,1,0,3,12]`

Output: `[1,3,12,0,0]`