Arrays

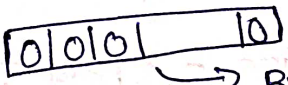
→ What are Arrays ?

int i=10;

int arr[] = new int[10];

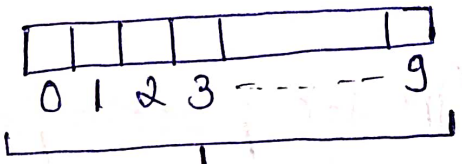
- [] → INDICATES This is an array
- 10 → size of array

Whenever an integer array is created, all elements are initialized to 0.



→ By default values in Array is zero

#
```
Public class ArrayUse {
    Public static void main(String[] args) {
        int[] arr = new int[10];
        arr[1] = 15;
        System.out.println(arr[1]);
    }
}
```



10 Integers

n → 0 --- (n-1)

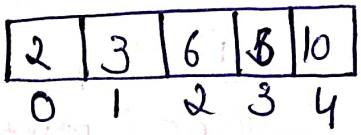
```
char chArray[] = new char[1];
chArray[0] = 'a';
double dArray[] = new double[12];
dArray[0] = 1.2;
System.out.println(chArray[0]);
System.out.println(dArray[0]);
```

whenever a boolean array is created, all elements are initialized to false.

Passing Arrays to function

```
import java.util.Scanner;
Public class ArrayUse{
  Public static void main(String[] args){
  Scanner s = new Scanner(System.In);
  Int size = s.nextInt();
  Int Input[] = new Int[size];
  for(int i=0; i<size; i++){
      Input[i] = s.nextInt();
    }

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



Make function for Program to return the Input element

for array

```
-> Public static Int[] take_Input(){
    Scanner s = new Scanner(System.In);
    int size = s.nextInt();
    input[] = new Int[size];
    for(int i=0; i<size; i++){
        Input[i] = s.nextInt();
    }
    return Input;
}
```

```
Public static void Print(int Input[]){
    int size = Input.length;        // —— FOR CALCULATING SIZE
    for(int j=0; j<size; j++){
        System.out.print(Input[j] + " ");
    }
    System.out.println();
}

Public static void main(String[] args){
    int arr[] = takeInput();
    Print(arr);
}
```

—— To calculate max from given elements

```
Public static int largestInArray(int Input[]){
    int max = Integer.MIN_VALUE;
    for(int j=0; j<Input.length; j++){
        if(Input[j] > max){
            max = Input[j];
        }
    }
    return max;
}
```

TO PUT ANY No equal to $-\infty$.

```
Public static void main(String[] args){
    int arr[] = takeInput();
    Print(arr);
    int largest = largestInArray(arr);
    System.out.println("Largest" + largest);
}
```

Array sum

Given an array/list (ARR) of length N, you need to find and return the sum of all elements in the array/list.

```
Public class solution {
    Public static int sum(int[] arr) {
        int sum = 0;
        int size = arr.length;
        for (int i=0; i<size; i++) {
            sum = sum + arr[i];
        }
        return sum;
    }
}
```

Linear search.

```
Public class solution {
    Public static int linearSearch (int arr[], int x) {
        int size = arr.length;
        for(int i=0; i<size; i++) {
            if(arr[i] == x)
                { return(i);
                }
            else { return(-1);
                }
```

Arrange Numbers in Array

```
Public class solution {
  Public static void arrange(int[] arr, int n) {
    int h=1;
    int J;
    if(n%2==0){
        J=n;
    }
    else {
        J=n-1;
    }
for(int i=0; i<arr.length; i++) {
  if(i<=(n-1)/2) {
      arr[i]=h;
      h++;
      h++;
  }
  else {
      arr[i]=J;
      J--;
      J--;
  }
}
}
```

```
{ int i=0, J=n-1
  int c=1;
  int l= arr.length;
for(c=1; c<=l; c++){
  if(c%2!=0){
  arr[i]=c;
   i++;
  }
  else {
   arr[J]=c;
    J--;
  }
}
}
```

1,3,5 ---, 6,4,2

	0	1	2	3	4	5
N=5	1	3	5	4	2	
N=6	1	3	5	6	4	2

N/2 = 5/2

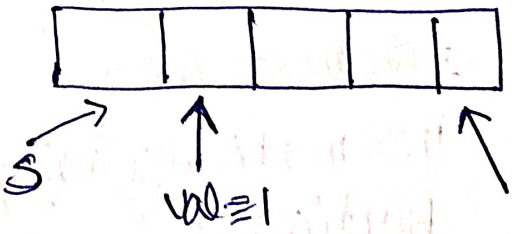
N/2 = ③

(N-1)/2 = ②

★

val
→ odd = N-1
→ even = N

[s] = val
val++
[e] = val
val++
s++
e--



How are Arrays Stored

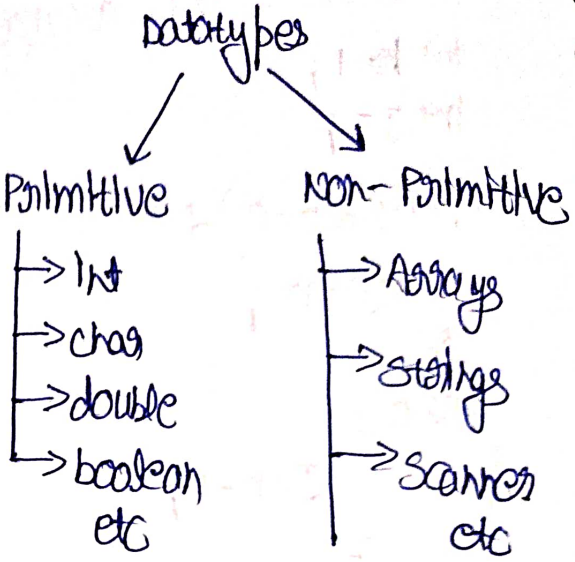
arr is reference of the array

— Address

```
{
    int arr[] = takeInput();
    System.out.println(arr);
}
```

OUTPUT

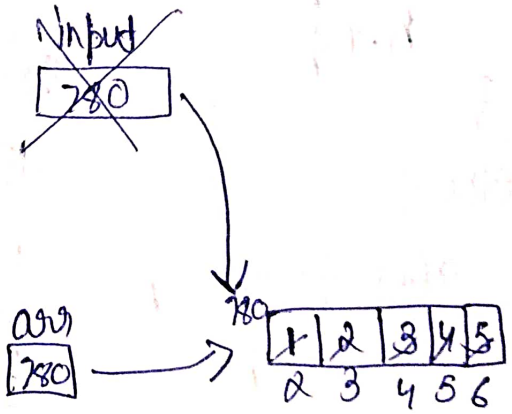
[I@4a57993



#

```
Public static void IncrementArray(int Input[]) {
    for(int i=0; i<Input.length; i++)
    {
        Input[i]++;
    }
}
```

→ Changes are deflected because they are shared on same memory location.




```
Public static void main (String[] args) {
    int arr[] = {1,2,3,4,5,};
    ~~increment array~~
    IncrementArray(arr);
    print(arr);
}
```

OUTPUT

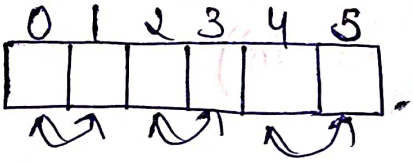
2,3,4,5,6

OUTPUT of the following CODE

```
Public class main{
    Public static void change(int input[]){
    input[0] = 15;
    }
Public static void main(string args[]){
    int arr[]= new int [5];
    change(arr);
    System.out.println(arr[0]);
}
}
```

OUTPUT
15

SWAP ALTERNATE




```
for(i=0 ; i<n-1; i++){
    c = arr[i]
    arr[i] = arr[i+1]
    arr[i+1] = c ;
} i++;
```

~~arr~~ ~~arr[0] =~~
c = arr[0] ;
arr[0] = arr[1] ;
arr[1] = c ;

```
{
  int c ;
  for(int i=0 ; i<arr.length-1 ; i++)
  { c = arr[i] ;
    arr[i] = arr[i+1];
    arr[i+1] = c;
    i++ ;
  }
}
```

Find Unique

#

```
Public static void main(String[] args) {
  int arr[]= {2,9,3,0,8,3,3,2,9,1};
  int n =10;
  for(int i=0; i<n ; i++) {
    int cnt = 0;
    for(int j=0; j<n ; j++) {
      if (arr[i]==arr[j]) cnt++;
    }
    if (cnt ==1) System.out.println(arr[i]);
  }
```

// Time complexity is O(n²) → two loops

FIND DUPLICATE

0	1	2	3	4
0	2	1	2	3

Arr(N)

N=5

↳ 0 to 3

2	3	0	2	3

(2N+1)

UNIQUE NUMBER

```
{
for(int i=0 ; i<arr.length ; i++){
  int count=0;
  for(int j=0 ; j<arr.length ;j++){
    if(arr[i]==arr[j] && i!=j){
        count++;
    }
  }
    if(count==0){
      return arr[i]; }
    }
    return(-1);
}
}
```

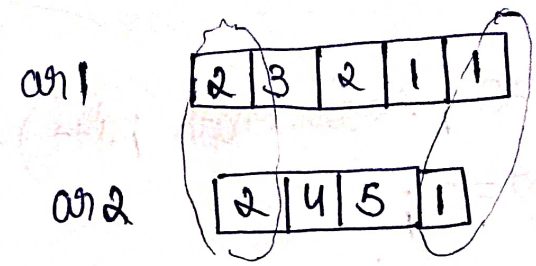
#

```
for(int i=0; i<arr.length ; i++){
  int count=0;
 for(int j=0 ; j<arr.length ;j++){
 if(arr[i]=arr[j] && i!=j){
      count++;
    }
 }
 if(count==1){
     return arr[i]; }
 }
  return(-1);
}
```

DUPLICATE NUMBER

~~Pair Sum~~

Array Intersection



↳ arr2 ma no. mil jaya ka bad usa $-\infty$ pet karo.

```
{
  for(int i=0 ; i<arr1.length ; i++) {
    for(int j=0 ; j<arr2.length ; j++) {
      if(arr1[i] == arr2[j]) {
        System.out.print(arr1[i] + " ");
        arr2[j] = Integer.MIN_VALUE ;
        break ;
      }
    }
  }
```

A1 : 6 2 3
A2 : 2 6 2 3 4

A1 : 2 6 2 3
A2 : 2 6

A1 : 2 6 2 3 2
A2 : 3 2 2

Pairs sum

Sum (7)

0	1	2	3	4	5	6	7
6	3	4	6	1	7	2	6

```
for(int i=0 ; i< arr.length ; i++)
    for(int j= i+1 ; j< arr.length ; j++)
    {  if(a[i] + a[j] == x)
        {  if(a[i] > a[j])
            {  a[j], a[i]
            }
           else
            {  a[i], a[j]
            }
        }
```

i=0 index → 1, 2, 3, (4), 5, 6, 7
i=1 → (2) 3, 4, 5, 6, 7
i=2 → 3, 4, 5, 6, 7
i=3 → 4,

END and return total no. of pairs in ~~which~~ the array/list sum to x.

~~**~~ — NHI AAYA

```
{
  int nextzero = 0;
  for(int i=0; i<arr.length ; i++)
  {  if(arr[i]==0)
     {  int temp = arr[nextzero];
        arr[nextzero] = arr[i];
        arr[i] = temp;
        nextzero++;
     }
  }
}
```

```
{
  int count = 0;
  for(int i=0 ; i<arr.length ; i++)
  {  int j=0;
     for(j=i+1 ; j<arr.length ; j++){
        if(arr[i] + arr[j] == x){
           count++;
        }
     }
  }
  return count;
}
```

SORT(0,1)