Arrays → Simplest Data structure

```
int arr[] = new int[5]
```

sequential collection of same type of data

```
int[]  arr  = new  int[5]
```

- int[] → datatype
- arr → name
- 5 → size of array

	0	1	2	3	4
arr →	10	20	30	40	50

print(arr[3]) 40

10 → 0 1 2 ---- 8 9

size = N 0, 1, 2, ----, n-1

3, 4, 1, 5, 1

①	②	③
Omoyh	Aman	Bharat
0	1	2

start = 100 + 2 + 0

H100 H101 H102 ----

Q Print sum of array

N = 5 {1, 2, 3, 4, 5}

sum = arr[0] + arr[1] + arr[2] + - - .

- - - - + arr[n-1]

First integer input is N
next is N integers which is the array itself

3 10 -6 5

```
int N = scn.nextInt()
int [] arr = new int[N]

for( i=0 ; i<N ; i++ ){
|       arr[i] = scn.nextInt()
}
int sum = 0
for (i=0 ; i <N ; i++) {
|       sum += arr[i]
}
  print (sum)
```

Q Print max elem of array

N = 5 { 3, 4, 1, 4, 1 }

```
int biggest = 0
for( i=0 ; i<n ; i++ ){
        if ( arr[i] > biggest )
                biggest = arr[i]
}
```

-3 -7 -2 -10 -1

```
int biggest = arr[0]
for( i=0 ; i<n ; i++ ){
        if ( arr[i] > biggest )
                biggest = arr[i]
}
```

Q Print min elem of array

N = 5 { 3, 4, 1, 4, 1 }

```
int smallest = arr[0]
for( i=0 ; i<n ; i++) {
        if ( arr[i] < smallest )
                smallest = arr[i]
}
```

Q Check if integer k is present in array or not

{1, 2, 3, 4, 5}	k = 2	true
{1, 2, 3, 4, 5}	k = 20	false

```
for( i=0; i<n; i++ ){
    if( arr[i] == k ){
        return true
    }
}
return false
```

Q. Return frequency of k in array

{1, 2, 3, 4, 5, 1} k = 1 ans = 2

```
int frequency (int N, int arr[], int k) {
    int count = 0
    for( i=0; i<n; i++ ) {
        if( arr[i] == k ) {
            count ++
        }
    }
    return count
}
```

Q Return frequency array for the input array.

{1, 1, 2, 1, 3, 1, 3}

ans = {4, 4, 1, 4, 2, 4, 2}

```
int[] freq_array (int arr[]){
    int n = arr.length
    int[] ans = new int[n]

    for( i=0; i<n; i++){
        ans[i] = frequency(n, arr, arr[i])
    }
    return ans
}
```

Q Check if array is strictly increasing or not

{1, 2, 3, 4, 5} true

{2, 4, 5, 1, 3} false

Idea: each element of the array should be bigger than previous elem

ar[1] > ar[0]
ar[2] > ar[1]
⋮
ar[n-1] > ar[n-2]

```
boolean increasing ( int [] arr) {

    int n = arr . length
    for( i=1 ; i<n ; i++){
        if ( ar [i] > ar [i-1])
            continue
        else
            return false
    }
    return true
}
```

[1, 7, 12, 14, 18]

arr[0] − arr[1] + arr[2] − arr[3] + arr[4]
− arr[5] + ...

```
sum=0
for( i=0 ; i<n ; i++ ){
    if ( i%2 == 0 )
        sum += arr[i]

    else
        sum -= arr[i]
}
```