

## 1-d Arrays

Ques) Take 5 integers as input & print their sum.

```
int a1, a2, a3, a4, a5;
```

```
// take input of all
```

```
// print their sum.
```

## Arrays

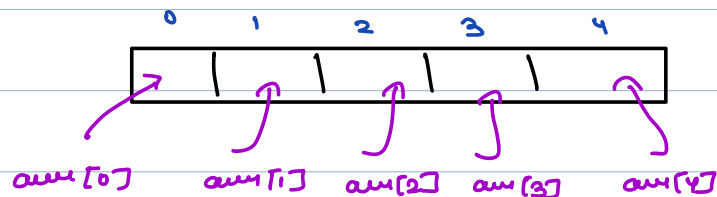
↳ Sequential collection of similar data

Train, YT playlist, movie list.

Syntax;

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

↓                      ↓                      ↓  
type                  variable                  size of array.



int  $\rightarrow$   $(-\infty, +\infty)$   
 $\underline{\underline{-109}}$

0	1	2	3	4
10	20	30	14	55

```

int max = 0; // Integer MIN-VALUE
for(int i = 0; i < n; i++){
    if(arr[i] > max){
        max = arr[i];
    }
}

System.out.println(max);

```

0	1	2	3	4
-10	-2	-8	-4	-6

Ques) Given an Integer array as input,  
 check whether  $k$  is present in array or not.

arr[]: { 3, 8, -1, 10, 0 }

$k = 8 \rightarrow \text{True}$

$k = 100 \rightarrow \text{false}$

$k = 0 \rightarrow \text{True}$

main() {

    print(findK(arr, k));  
 }

boolean findK (int[] arr, int k) {

for (int i = 0; i < arr.length; i++) {

if (arr[i] == k) {

return true;

return false;

arr[]: { 3, 8, -1, 10, 0 } , k = 8 ,  
k = 5

Ques)

Given an Integer array, return freq of  $k$  in the array.

arr = { 1, 2, 1, 1, 3, 0, 2, 1, 2 }

$k = 1 \rightarrow 4$

$k = 2 \rightarrow 3$

$k = 5 \rightarrow 0$

```
int frequencyK ( int[] arr, int k ) {
```

```
    int cnt = 0;
```

```
    for ( i = 0; i < arr.length; i++ ) {
```

```
        if ( arr[i] == k ) {
```

```
            cnt++;
```

```
        }
```

```
    }
```

```
    return cnt;
```

Ques Given an integer array, return  
freq. count of array.

arr[] = { 1, 1, 2, 1, 3, 1, 3 }

new arr[] = { 4, 4, 1, 4, 2, 4, 2 }

int[] freqCount ( int [] arr ) {

int[] newArr = new int[arr.length];  
for ( i = 0; i < <sup>arr.length</sup> ; i++ ) {

newArr[i] = frequencyK(arr, arr[i]);

return newArr;

Main ( )

→ arr  
int[] ans = freqCount(arr);

Ques) Given an integer array, check whether it is strictly increasing.

arr[] = { 0, 2, 6, 7, 8 } → True.

arr[] = { 0, 2, 2, 6, 7, 8 } → false

arr[] = { 0, -5, 2, 3, -4 } → false.

arr → 5 → 0 to 4

bool isIncreasing (int arr[]) {

int n = arr.length;

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

if (arr[i] >= arr[i+1]) {

return false

return true;



boolean isPrime (n) {

int cnt = 0;

for (i = 1; i <= n; i++) {

if (n % i == 0) {

cnt++

}

if (cnt == 2) {

return true;

}

else {

return false;

}

}

1 to 20

function getAllPrimes (n) {

for (i = 2; i <= n; i++) {

if (isPrime(i)) {

print(i);

}

}

}