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

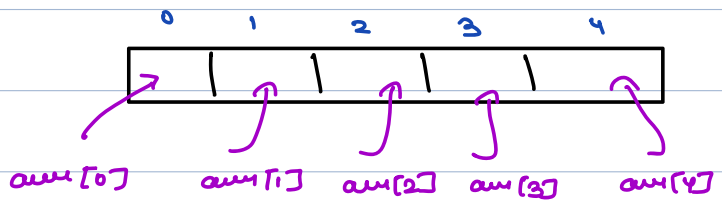
↳ Sequential collection of similar data

Train, YT playlist, movie list.

Syntax;

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

- `int []` ↓ type
- `arr` ↓ variable name
- `new int[5]` ↓ size of array.



int → $(-\infty, +\infty)$ → -2^{31} to -10^9

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

System.out.println(max);
```

Integer.MIN_VALUE

0	1	2	3	4
10	20	300	14	55

0	1	2	3	4
-1	-2	-3	-4	-5

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

arr[7] = { 3, 8, -1, 10, 0 }

k = 8 → True

k = 100 → false

k = 0 → 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 arr, return freq of k in the array.

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

k = 1 ⟶ 4

k = 2 ⟶ 3

k = 5 ⟶ 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< arr.length ; i++) {
        newArr[i] = frequency(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, 7} → false

arr[] = { 0, -5, 2, 3, -4 } → false.

arr → 5 → 0 to 4

```
boolean isIncreasing ( int arr[] ) {
    int n = arr.length;
    for (i=0; i<n-1 ; i++) {
        if (arr[i] >= arr[i+1]) {
            return false
        }
    }
    return true;
}
```

https://www.scaler.com/topics/java/online-java-compiler/?snippet_slug=843bbba29fb37f6d84ec

```
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 200

```
function getAllPrimes (n) {
    for (i = 2; i <= n; i++) {
        if (isPrime(i)) {
            print (i);
        }
    }
}
```