Check if the Array is Sorted →

* non-descending array (ascending / Increasing):

i) if current element is greater than or equal to previous element, then return ~~false~~ true.

ii) ~~else return false.~~

i) If any current element is smaller than the previous, then array is not sorted → return false

ii) If no such case is found, the array is sorted, return true.

Sol^n :→

Start from index 1

```
for (int i=1 ; i<n ; i++)
{
  if (arr[i] < arr[i-1]) → (i)
  {
    return false
  }
}
return true //otherwise → (ii)
```

Dry Run :→ arr = {1, 2, 3, 4, 5}, n = 5

i = 1	i = 2	i = 3	i = 4	i = 5
arr[i] = 2	arr[2] = 3	arr[3] = 4	arr[4] = 5	↓
arr[i-1] =	arr[1] = 2	arr[2] = 3	arr[3] = 4	exit for loop
arr[0] = 1	3 < 2 → X	4 < 3 → X	5 < 4 → X	↓
2 < 1 → X	continue	continue	continue	return true.
continue				

Dry Run:→ ② arr = {1, 3, 2, 4, 5}

arr[i] = arr[1] = 3
arr[i-1] = arr[0] = 1
3 < 1 → ×
continue

arr[i] = arr[2] = 2
arr[i-1] = arr[1] = 3
2 < 3 → condition matched.
return false