

IsSorted in C++																																					
<pre>#include &lt;iostream&gt; using namespace std;  bool isSortedEff(int arr[], int n) {     for (int i = 1; i &lt; n; i++) {         if (arr[i] &lt; arr[i - 1]) {             return false;         }     }     return true; }  bool isSorted(int arr[], int n) {     for (int i = 0; i &lt; n; i++) {         for (int j = i + 1; j &lt; n; j++) {             if (arr[j] &lt; arr[i]) {                 return false;             }         }     }     return true; }  int main() {     int arr1[] = {1, 2, 3, 4, 5, 6};     int arr2[] = {11, 2, 3, 4, 5, 6};     int n1 = sizeof(arr1) / sizeof(arr1[0]);     int n2 = sizeof(arr2) / sizeof(arr2[0]);      cout &lt;&lt; boolalpha; // Print boolean values as true/false     cout &lt;&lt; isSortedEff(arr1, n1) &lt;&lt; endl;     cout &lt;&lt; isSortedEff(arr2, n2) &lt;&lt; endl;      cout &lt;&lt; isSorted(arr1, n1) &lt;&lt; endl;     cout &lt;&lt; isSorted(arr2, n2) &lt;&lt; endl;      return 0; }</pre>		Check if an array is <b>sorted in non-decreasing order</b> (each element is $\leq$ the next).																																			
		🔍 Difference between isSortedEff and isSorted:																																			
		<table><tr><th>Function</th><th>Approach</th><th>Time Complexity</th></tr><tr><td>isSortedEff</td><td>Linear scan (compare adjacent)</td><td><b>O(n)</b></td></tr><tr><td>isSorted</td><td>Brute force (nested loops)</td><td><b>O(n²)</b></td></tr></table>			Function	Approach	Time Complexity	isSortedEff	Linear scan (compare adjacent)	<b>O(n)</b>	isSorted	Brute force (nested loops)	<b>O(n²)</b>																								
		Function	Approach	Time Complexity																																	
		isSortedEff	Linear scan (compare adjacent)	<b>O(n)</b>																																	
		isSorted	Brute force (nested loops)	<b>O(n²)</b>																																	
		✔ Dry Run with Sample Arrays																																			
		Array 1: {1, 2, 3, 4, 5, 6} (Sorted)																																			
		isSortedEff(arr1, n1):																																			
		<table><tr><th>i</th><th>arr[i-1]</th><th>arr[i]</th><th>Comparison</th><th>Result</th></tr><tr><td>1</td><td>1</td><td>2</td><td><math>2 \geq 1</math></td><td>✔</td></tr><tr><td>2</td><td>2</td><td>3</td><td><math>3 \geq 2</math></td><td>✔</td></tr><tr><td>3</td><td>3</td><td>4</td><td><math>4 \geq 3</math></td><td>✔</td></tr><tr><td>4</td><td>4</td><td>5</td><td><math>5 \geq 4</math></td><td>✔</td></tr><tr><td>5</td><td>5</td><td>6</td><td><math>6 \geq 5</math></td><td>✔</td></tr><tr><td colspan="5">→ All passed → <b>Returns: true</b></td></tr></table>			i	arr[i-1]	arr[i]	Comparison	Result	1	1	2	$2 \geq 1$	✔	2	2	3	$3 \geq 2$	✔	3	3	4	$4 \geq 3$	✔	4	4	5	$5 \geq 4$	✔	5	5	6	$6 \geq 5$	✔	→ All passed → <b>Returns: true</b>		
i	arr[i-1]	arr[i]	Comparison	Result																																	
1	1	2	$2 \geq 1$	✔																																	
2	2	3	$3 \geq 2$	✔																																	
3	3	4	$4 \geq 3$	✔																																	
4	4	5	$5 \geq 4$	✔																																	
5	5	6	$6 \geq 5$	✔																																	
→ All passed → <b>Returns: true</b>																																					
isSorted(arr1, n1): Checks every pair (i, j) where $j > i$ :																																					
<ul style="list-style-type: none"><li>For every <math>\text{arr}[i] \leq \text{arr}[j] \rightarrow</math> all OK <math>\rightarrow</math> <b>Returns: true</b></li></ul>																																					
Array 2: {11, 2, 3, 4, 5, 6} (Not sorted)																																					
isSortedEff(arr2, n2):																																					
<table><tr><th>i</th><th>arr[i-1]</th><th>arr[i]</th><th>Comparison</th><th>Result</th></tr><tr><td>1</td><td>11</td><td>2</td><td><math>2 &lt; 11</math> ✖</td><td>●</td></tr><tr><td colspan="5">→ Early exit → <b>Returns:</b></td></tr></table>			i	arr[i-1]	arr[i]	Comparison	Result	1	11	2	$2 < 11$ ✖	●	→ Early exit → <b>Returns:</b>																								
i	arr[i-1]	arr[i]	Comparison	Result																																	
1	11	2	$2 < 11$ ✖	●																																	
→ Early exit → <b>Returns:</b>																																					

	<table><tr><th>i</th><th>arr[i-1]</th><th>arr[i]</th><th>Comparison</th><th>Result</th></tr><tr><td>false</td><td></td><td></td><td></td><td></td></tr></table>	i	arr[i-1]	arr[i]	Comparison	Result	false				
i	arr[i-1]	arr[i]	Comparison	Result							
false											
	<p>isSorted(arr2, n2):</p> <ul style="list-style-type: none"><li>(0,1) → 2 &lt; 11 → ✖ → Returns: false</li></ul> <p>🖨 Output:</p> <p>true false true false</p>										
true false true false											