Bubble sort

for(int i=0;i< arr.length-1;i++){  
 for(int j=0;j<arr.length-1-i;j++){ // once the element sorted not that position we need to exclude it  
 if(arr[j]>arr[j+1]){  
 int temp=arr[j];  
 arr[j]=arr[j+1];  
 arr[j+1]=temp;  
 }  
 }  
  
}

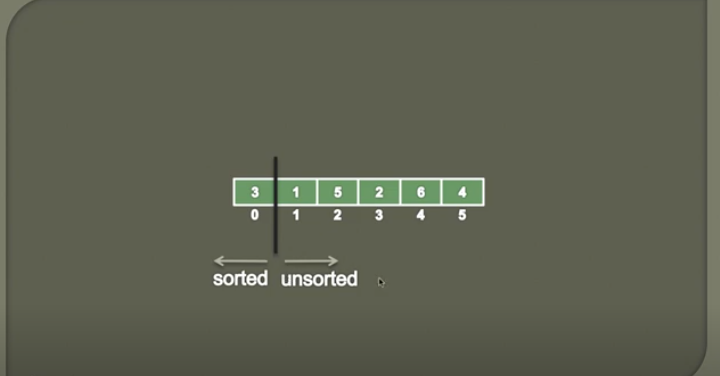
2. Insertion Sort:

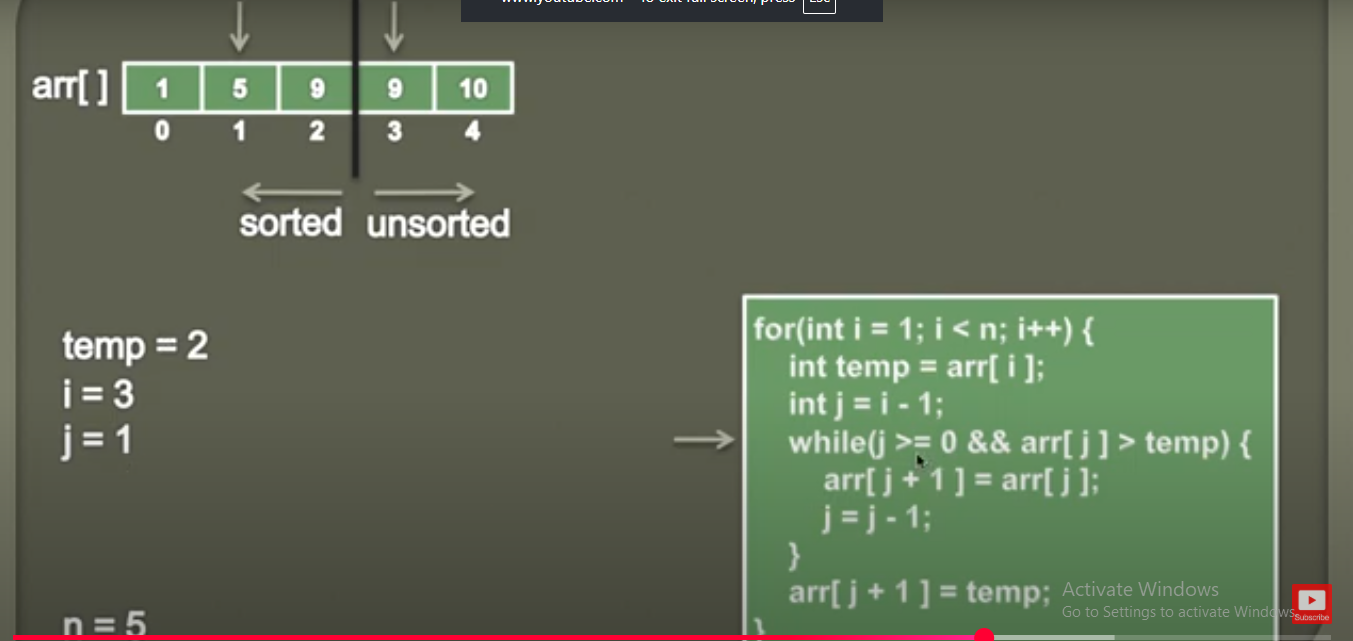
**Scenario:**

Imagine you are organizing books on a shelf by **height** (shortest to tallest).  
You're handed **one book at a time**, and you insert it into the correct position among the already-placed books.

**Step-by-Step Analogy:**

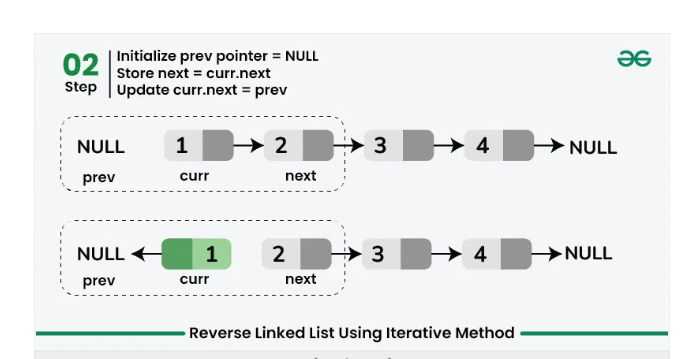
1. You place the **first book** — no problem, it’s the only one. ✅
2. The **second book** arrives. You compare it to the first:
   * Is it shorter? Put it before.
   * Is it taller? Put it after.
3. A third book arrives. Compare it with the books already on the shelf.
4. You continue **inserting each new book into the right spot**.





int a[]={9,2,89,8,7,6,65,4,3,21};  
  
for (int i=1;i<a.length;i++){  
 int temp=a[i];  
 int j=i-1;  
 while(j>0 && a[j]>temp){  
 a[j+1]=a[j];  
 j--;  
 }  
a[j+1]=temp;  
}

LInkedList

By seeing below pic we can understand the scenario,   


Node curr = head, prev = null, next;

while (curr != null) {

next = curr.next;

// Reverse current node's next pointer

curr.next = prev;

// Move pointers one position ahead

prev = curr;

curr = next;

}

// Return the head of reversed linked list

return prev;