**SORTING ALGORITHMS IN C**

**Bubble Sort**

[Bubble Sort](https://www.edureka.co/blog/bubble-sort-in-c/) is a simple sorting algorithm which repeatedly compares the adjacent elements of the given array & swaps them if they are in wrong order.  
Suppose we have an array X which contains n elements which needs to be sorted using Bubble Sort. The sorting works as:

**Pass 1:**

* X[0] & X[1] are compared, and swapped if X[0] > X[1]
* X[1] & X[2] are compared, and swapped if X[1] > X[2]
* X[2] & X[3] are compared, and swapped if X[2] > X[3] and so on…

At the end of pass 1, the largest element of the list is placed at the highest index of the list.

**Pass 2:**

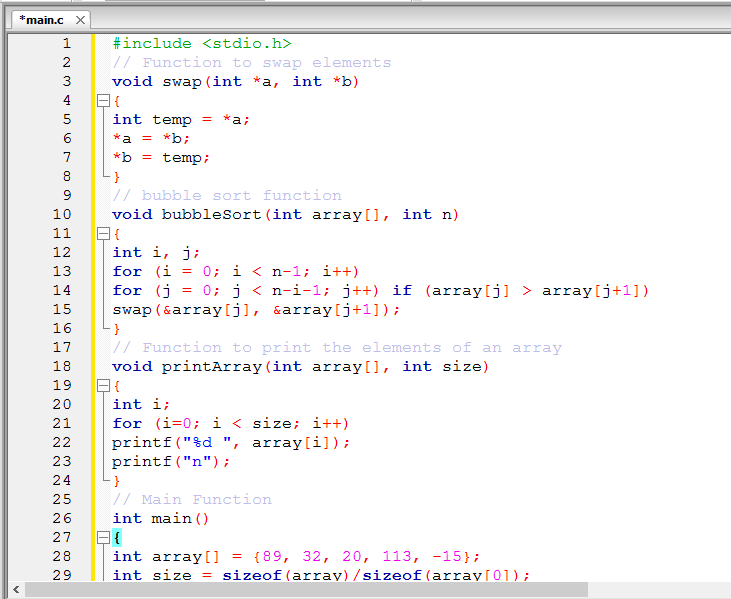
* X[0] & X[1] are compared, and swapped if X[0] > X[1]
* X[1] & X[2] are compared, and swapped if X[1] > X[2]
* X[2] & X[3] are compared, and swapped if X[2] > X[3] and so on…

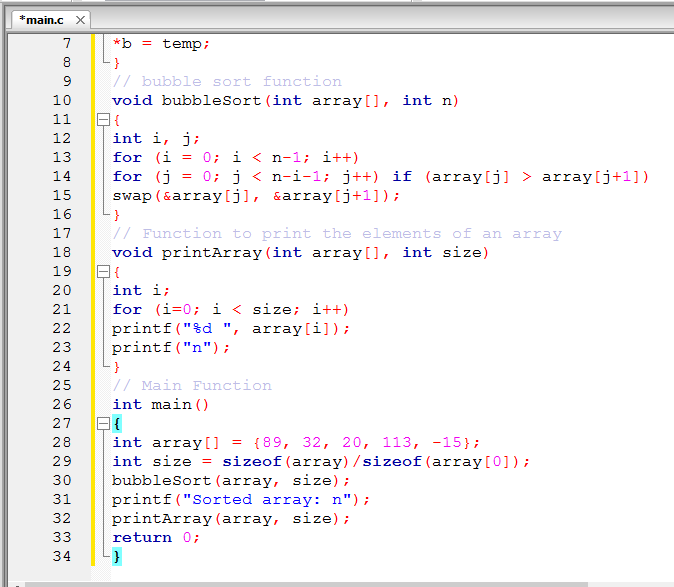
At the end of Pass 2 the second largest element of the list is placed at the second highest index of the list.

**Pass n-1:**

* X[0] & X[1] are compared, and swapped if X[0] > X[1]
* X[1] & X[2] are compared, and swapped if X[1] > X[2]
* X[2] & X[3] are compared, and swapped if X[2] > X[3] and so on…

At the end of this pass, the smallest element of the list is placed at the first index of the list.

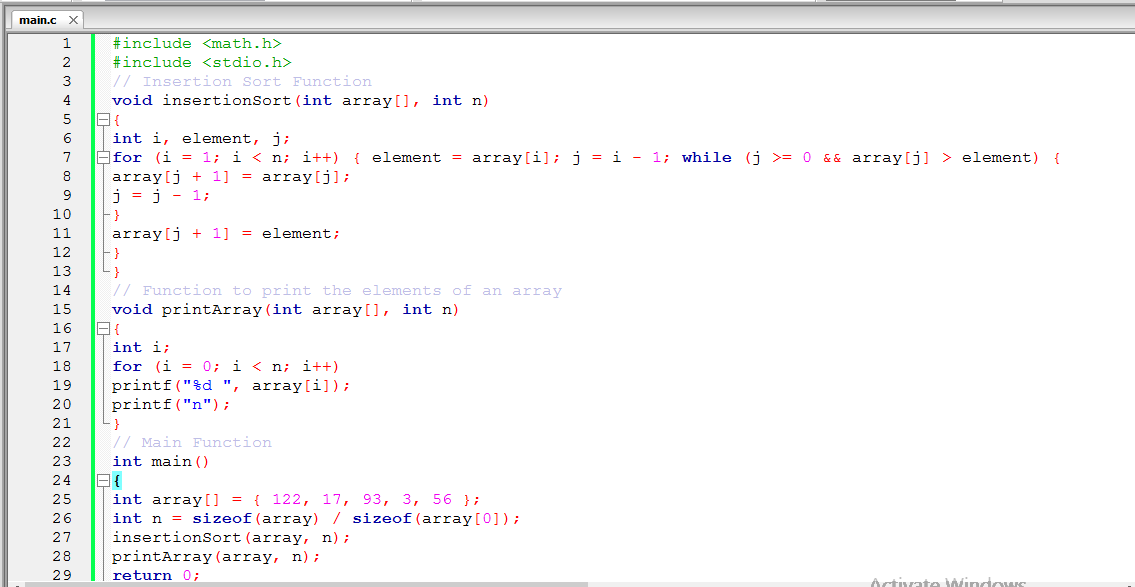




## ****Insertion Sort****

[Insertion Sort](https://www.edureka.co/blog/insertion-sort-in-c/) is a sorting algorithm where the array is sorted by taking one element at a time. The principle behind insertion sort is to take one element, iterate through the sorted array & find its correct position in the sorted array.

**Step 1** − If the element is the first one, it is already sorted.  
**Step 2** – Move to next element  
**Step 3** − Compare the current element with all elements in the sorted array  
**Step 4** – If the element in the sorted array is smaller than the current element, iterate to the next element. Otherwise, shift all the greater element in the array by one position towards right  
**Step 5** − Insert the value at the correct position  
**Step 6** − Repeat until the complete list is sorted



## ****Selection Sort in C****

Selection sort is another algorithm that is used for sorting. This sorting algorithm, iterates through the array and finds the smallest number in the array and swaps it with the first element if it is smaller than the first element. Next, it goes on to the second element and so on until all elements are sorted.

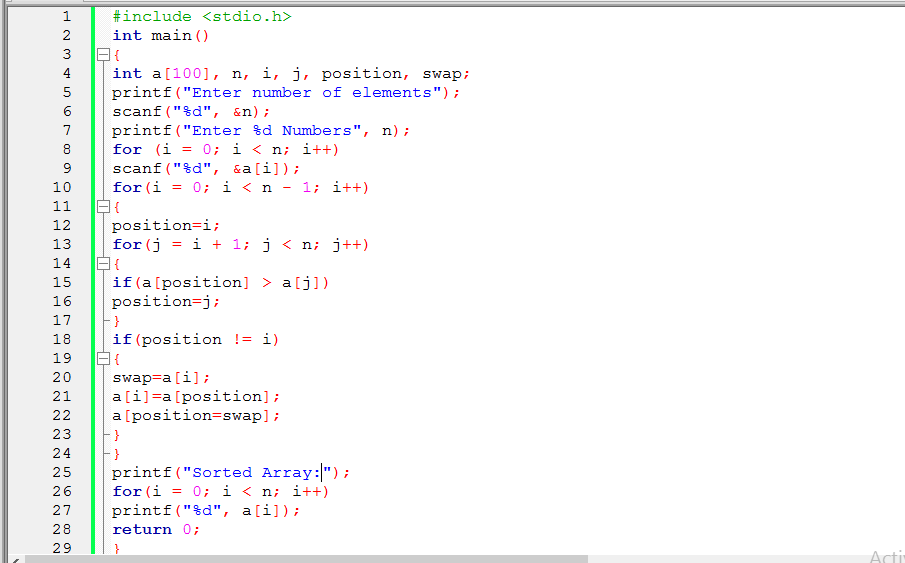
**Step 1** − Set min to the first location

**Step 2** − Search the minimum element in the array

**Step 3** – swap the first location with the minimum value in the array

**Step 4** – assign the second element as min.

**Step 5** − Repeat the process until we get a sorted array.



## ****Merge Sort****

[Merge Sort](https://www.edureka.co/blog/merge-sort-in-c/) is one of the best examples of Divide & Conquer algorithm. In Merge sort, we divide the array recursively in two halves, until each sub-array contains a single element, and then we merge the sub-array in a way that it results into a sorted array. merge() function merges two sorted sub-arrays into one, wherein it assumes that array[l .. n] and arr[n+1 .. r] are sorted.

