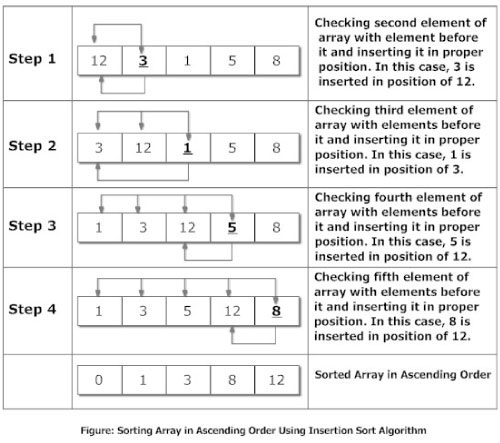
**How insertion sort Algorithm works?**



**Explanation**

**Suppose, you want to sort elements in ascending as in above figure. Then,**

1. **Step 1: The second element of an array is compared with the elements that appears before it (only first element in this case). If the second element is smaller than first element, second element is inserted in the position of first element. After first step, first two elements of an array will be sorted.**
2. **Step 2: The third element of an array is compared with the elements that appears before it (first and second element). If third element is smaller than first element, it is inserted in the position of first element. If third element is larger than first element but, smaller than second element, it is inserted in the position of second element. If third element is larger than both the elements, it is kept in the position as it is. After second step, first three elements of an array will be sorted.**
3. **Step 3: Similarly, the fourth element of an array is compared with the elements that appears before it (first, second and third element) and the same procedure is applied and that element is inserted in the proper position. After third step, first four elements of an array will be sorted.**

**If there are *n* elements to be sorted. Then, this procedure is repeated *n-1* times to get sorted list of array.**

**Insertion Sort Program**

**#include<stdio.h>**

**int main()**

**{**

**int array[25],n,temp,i,j;**

**printf("Enter number of terms ");**

**scanf("%d",&n);**

**printf("Enter elements: ");**

**for(i=0;i<n;i++)**

**{**

**scanf("%d",&array[i]);**

**}**

**for(i=1;i<n;i++)**

**{**

**temp = array[i];**

**j=i-1;**

**while(temp<array[j] && j>=0)**

**/\*To sort elements in descending order, change temp<data[j] to temp>data[j] in above line.\*/**

**{**

**array[j+1] = array[j];**

**--j;**

**}**

**array[j+1]=temp;**

**}**

**printf("In ascending order: ");**

**for(i=0; i<n; i++)**

**printf("%d\t",array[i]);**

**return 0;**

**}**

**Selection sort algorithm** starts by comparing first two elements of an array and swapping if necessary, i.e., if you want to sort the elements of array in ascending order and if the first element is greater than second then, you need to swap the elements but, if the first element is smaller than second, leave the elements as it is. Then, again first element and third element are compared and swapped if necessary. This process goes on until first and last element of an array is compared. This completes the first step of selection sort.

If there are *n* elements to be sorted then, the process mentioned above should be repeated *n-1* times to get required result. But, for better performance, in second step, comparison starts from second element because after first step, the required number is automatically placed at the first (i.e, In case of sorting in ascending order, smallest element will be at first and in case of sorting in descending order, largest element will be at first.). Similarly, in third step, comparison starts from third element and so on.



**#include <stdio.h>**

**int main()**

**{**

**int data[100],i,n,steps,temp;**

**printf("Enter the number of elements to be sorted: ");**

**scanf("%d",&n);**

**for(i=0;i<n;++i)**

**{**

**printf("%d. Enter element: ",i+1);**

**scanf("%d",&data[i]);**

**}**

**for(steps=0;steps<n;++steps)**

**for(i=steps+1;i<n;++i)**

**{**

**if(data[steps]>data[i])**

**/\* To sort in descending order, change > to <. \*/**

**{**

**temp=data[steps];**

**data[steps]=data[i];**

**data[i]=temp;**

**}**

**}**

**printf("In ascending order: ");**

**for(i=0;i<n;++i)**

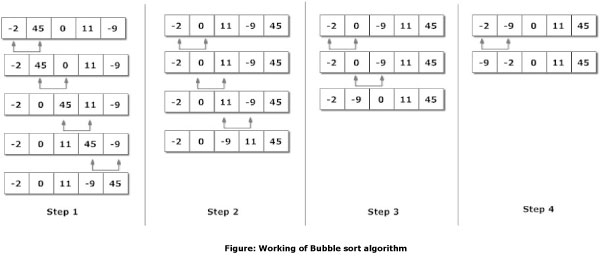
**printf("%d ",data[i]);**

**return 0;**

**}**

**Bubble sort algorithm starts by comparing the first two elements of an array and swapping if necessary, i.e., if you want to sort the elements of array in ascending order and if the first element is greater than second then, you need to swap the elements but, if the first element is smaller than second, you mustn't swap the element. Then, again second and third elements are compared and swapped if it is necessary and this process go on until last and second last element is compared and swapped. This completes the first step of bubble sort.**

**If there are *n* elements to be sorted then, the process mentioned above should be repeated *n-1* times to get required result. But, for better performance, in second step, last and second last elements are not compared because, the proper element is automatically placed at last after first step. Similarly, in third step, last and second last and second last and third last elements are not compared and so on.**

****

**/\* C program to sort N numbers in ascending order using Bubble sort and print both the given and the sorted array \*/**

#include <stdio.h>

void main()

{

int array[10];

int i, j, num, temp;

printf("Enter the value of num \n");

scanf("%d", &num);

printf("Enter the elements one by one \n");

for (i = 0; i < num; i++)

{

scanf("%d", &array[i]);

}

printf("Input array is \n");

for (i = 0; i < num; i++)

{

printf("%d\n", array[i]);

}

/\* Bubble sorting begins \*/

for (i = 0; i < num; i++)

{

for (j = 0; j < (num - i - 1); j++)

{

if (array[j] > array[j + 1])

{

temp = array[j];

array[j] = array[j + 1];

array[j + 1] = temp;

}

}

}

printf("Sorted array is...\n");

for (i = 0; i < num; i++)

{

printf("%d\n", array[i]);

}

}