

Insertion Sort

Insertion sort is a simple sorting algorithm: a comparison sort in which the sorted array (or list) is built one entry at a time. It is much less efficient on large lists than more advanced algorithms such as quicksort, heapsort, or merge sort.

Every repetition of insertion sort removes an element from the input data, inserting it into the correct position in the already-sorted list, until no input elements remain. The choice of which element to remove from the input is arbitrary, and can be made using almost any choice algorithm.

Sorting is typically done in-place. The resulting array after k iterations has the property where the first $k + 1$ entries are sorted. In each iteration the first remaining entry of the input is removed, inserted into the result at the correct position, thus extending the result:

Sorted partial result		Unsorted data	
$\leq x$	$> x$	x	...

becomes

Sorted partial result		Unsorted data	
$\leq x$	x	$> x$...

with each element greater than x copied to the right as it is compared against x .

CODES{C}

```
#include<stdio.h>
int main(){

    int i,j,s,temp,a[20];

    printf("Enter total elements: ");
    scanf("%d",&s);

    printf("Enter %d elements: ",s);
    for(i=0;i<s;i++)
        scanf("%d",&a[i]);

    for(i=1;i<s;i++){
        temp=a[i];
        j=i-1;
        while((temp<a[j])&&(j>=0)){
            a[j+1]=a[j];
            j=j-1;
        }
        a[j+1]=temp;
    }

    printf("After sorting: ");
    for(i=0;i<s;i++)
        printf(" %d",a[i]);

    return 0;
}
```

CODES{JAVA}

```
public class InsertionSort{
    public static void main(String a[]){
        int i;
        int array[] = {12,9,4,99,120,1,3,10};
        System.out.println("\n\n RoseIndia\n\n");
        System.out.println(" Selection Sort\n\n");
        System.out.println("Values Before the sort:\n");
        for(i = 0; i < array.length; i++)
            System.out.print( array[i]+" ");
        System.out.println();
        insertion_srt(array, array.length);
        System.out.print("Values after the sort:\n");
        for(i = 0; i < array.length; i++)
            System.out.print(array[i]+" ");
        System.out.println();
        System.out.println("PAUSE");
    }

    public static void insertion_srt(int array[], int n){
        for (int i = 1; i < n; i++){
            int j = i;
            int B = array[i];
            while ((j > 0) && (array[j-1] > B)){
                array[j] = array[j-1];
                j--;
            }
            array[j] = B;
        }
    }
}
```