INSERTION SORT -

The Insertion sort inserts each element in proper place. Suppose an array a with n elements a[0], a[1], a[2], .. a[n] is in memory. The insertion sort algorithm scans a from a[0] to a[~~n~~ n-1], inserting each element a[K] into its proper position in the previously sorted subarray a[0], a[1], a[2], ..., a[K-1]. That is.

Pass1 a[0] is trivially sorted.

Pass 2 a[1] is inserted either before or after a[0] so that a[0], a[1] is sorted.

Pass3 . . . .

. . . . inserted in its proper

Pass N a[n-1] is inserted in its proper place in a[0], a[1], a[2] . . . . . . a[n-2] so that the array is sorted.

The element inserted in the proper place is compared with the previous elements and placed in between the ith element & i+1 th element if

element >= ith element.

element <= (i+1) th element

Let us take the elements -

82 42 49 8 92 25 59 52

Pass1 - 82 42 49 8 92 25 59 52

Pass2 - 82 42 49 8 92 25 59 52

42 82 49 8 92 25 59 52

elementsalnka [i]

```
Pass 3   42   82   49   8   92   25   59   52
                        8   92   25   59   52
         42   49   82
                                 25   54   52
Pass 4   42   49   82   8   92   25   54   52
                                 92   95   54   52
Pass5    8    42   49   82  92
                                      54   52
Pass 6   8    25   42   49  82   92
                                      92   52
Pass 7   8    25   42   49  54   82
                            54   82   92
Pass 8   8    25   42   49  52   54   82   92
```

ALGORITHM- InsertionSort (A[ ], Item)

let a be an array of n elements which we want to sort,

```
1.  Set K=1
2   for K=1 to (n-1)
    Set temp =a[K]
    set j =k-1
    While temp < a[j] and (j>=0) perform
    fhe following steps.
    Set a[j+1]= a[j]
    [End of loop structure]
3   Exit
```

C Function -

```
void insertion -sort (int a[ ], int n)
{
 int i, K, j, temp;
for (K=1; K<=n-1; k++)
{ temp = a[K];
  j=k-1;
  while (temp < a[j] && (j>=0))
  {
        a[j+1]=a[j];
        j=j-1;
      }
  a[j+1]=temp;
}
printf("Elements of array after sorting are:\n");
~~for(i~~
  }
```