2. Problem Statement: Insertion Sort

**Problem Analysis:**

Insertion sort algorithm iterates, consuming one input element each repetition, and growing a sorted output list. Each iteration removes one element from the input data, finds the location it belongs within the sorted list, and inserts it there. It repeats until no input elements remain.

The insertion sort algorithm is performed using following steps:

*1.* Assume that first element in the list is in sorted portion of the list and remaining all elements are in unsorted portion.

*2.* Consider first element from the unsorted list and insert that element into the sorted list in order specified.

*3.* Repeat the above process until all the elements from the unsorted list are moved into the sorted list.

**Algorithm:**

InsertionSort(A,n)

*//Sort the array A[1:n] into non-decreasing order.*

{

for i:= 1 to n do;

{

value: = A[i];

k: = i;

while(k>0 and A[k-1]>value)

{

A[k]: = A[k-1];

k: = k-1;

}

A[k]: = value;

}

}

**Source Code:**

#include<iostream>

using namespace std;

void InsertionSort(int A[], int n)

{

int i,value,k;

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

{

value = A[i];

k = i;

while(k>0 && A[k-1]>value)

{

A[k] = A[k-1];

k = k-1;

}

A[k] = value;

}

}

int main()

{

int i,n,A[30];

cout<<"Enter array size: \n";

cin>>n;

cout<<"Enter array elements: \n";

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

cin>>A[i];

InsertionSort(A,n);

cout<<"Sorted array elements: \n";

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

cout<<A[i]<<" ";

return 0;

}

**Sample Input:**

Enter array size:

7

Enter array elements:

99 55 11 88 22 77 33

**Sample Output:**

Sorted array elements:

11 22 33 55 77 88 99