**ITERATIVE SOLUTIONS**

**BUBBLE SORT:**

#include<stdio.h>

int main()

{

    int n;

    scanf("%d", &n);

    int i = 0,array[n];

    for ( i = 0; i < n;i++)

    {

        scanf("%d", &array[i]);

    }

    int flag = 1;

    while (flag)

    {

        flag = 0;

        for (i = 0; i < n;i++)

        {

            if(array[i]>array[i+1])

            {

                flag = 1;

                int temp = array[i];

                array[i] = array[i + 1];

                array[i + 1] = temp;

            }

        }

    }

    for ( i = 0; i < n; i++)

    {

        printf("%d ", array[i]);

    }

}

**INSERTION SORT:**

#include <stdio.h>

int main()

{

    int n;

    scanf ("%d", &n);

    int i = 0, arr[n];

    for (i = 0; i < n; i++)

    {

        scanf ("%d", &arr[i]);

    }

    int key, j;

    for (i = 1; i < n; i++)

    {

        key = arr[i];

        j = i - 1;

        while (j >= 0 && arr[j] > key)

        {

            arr[j + 1] = arr[j];

            j = j - 1;

        }

        arr[j + 1] = key;

    }

    for (i = 0; i < n; i++)

    {

        printf("%d ", arr[i]);

    }

}

**SHELL SORT (OR) DIMINISHING DISTANCE SORT:**

#include <iostream>

using namespace std;

int shellSort(int arr[], int n)

{

    for (int gap = n / 2; gap > 0; gap /= 2)

    {

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

        {

            int temp = arr[i];

            int j;

            for (j = i; j >= gap && arr[j - gap] > temp; j -= gap)

                arr[j] = arr[j - gap];

            arr[j] = temp;

        }

    }

    return 0;

}

int main()

{

    int n;

    cin >> n;

    int arr[n];

    for (int i = 0; i < n; i++)

        cin >> arr[i];

    shellSort(arr, n);

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    return 0;

}