**RECURSIVE SOLUTIONS**

**BUBBLE SORT:**

#include <stdio.h>

int\* bubblesort(int a[],int flag,int n,int iterator)

{

    int it = iterator % n-1, it1 = it+1;

    if(iterator==n\*n)

        return a;

    if(a[it]>a[it1])

    {

        int temp = a[it1];

        a[it1] = a[it];

        a[it] = temp;

    }

    return bubblesort(a, flag+1, n, iterator + 1);

}

int main()

{

    int n;

    scanf("%d", &n);

    int i = 0, array[n];

    for (i = 0; i < n; i++)

    {

        scanf("%d", &array[i]);

    }

    int \*arr = bubblesort(array, 1, n, 0);

    for (i = 0; i < n; i++)

    {

       printf("%d ", arr[i]);

    }

    return 0;

}

**INSERTION SORT:**

#include <stdio.h>

int\* insetionSort(int arr[],int j,int key)

{

    if(j<0 && arr[j]<key)

    {

        arr[j + 1] = key;

        return arr;

    }

    arr[j + 1] = arr[j];

    return insetionSort(arr, j - 1, key);

}

int\* getInserSort(int arr[],int n)

{

    int i,key, j;

    for (i = 1; i < n; i++)

    {

        key = arr[i];

        j = i - 1;

        arr = insetionSort(arr, j, key);

    }

    return arr;

}

int main()

{

    int n;

    scanf("%d", &n);

    int i = 0, arr[n];

    for (i = 0; i < n; i++)

    {

        scanf("%d", &arr[i]);

    }

    int \*array = getInserSort(arr,n);

    for (i = 0; i < n; i++)

    {

        printf("%d ", array[i]);

    }

}

**MERGE SORT:**

#include<bits/stdc++.h>

using namespace std;

int\* merge(int arr[],int low,int high,int piv)

{

    int a[high-low+1];

    int x=low,y=piv+1,i=0;

    while(x<=piv && y<= high)

    {

        if(arr[x]<arr[y])

        {

            a[i++]=arr[x++];

        }

        else

        {

            a[i++]=arr[y++];

        }

    }

    while(x<=piv)

    {

        a[i++]=a[x++];

    }

    while(y<=high)

    {

        a[i++]=a[y++];

    }

    return a;

}

int\* mergeSort(int arr[],int low,int high)

{

    int piv=(low+high)/2;

    if(low<high)

    {

        arr=mergeSort(arr,low,piv);

        arr=mergeSort(arr,piv+1,high);

        arr=merge(arr,low,high,piv);

    }

    return arr;

}

int main()

{

    int n;

    int a[100],i=0;

    while((scanf("%d",&n))!=-1)

    {

        a[i++]=n;

    }

    int \*arr=mergeSort(a,0,i);

    for(int x=0;x<i;x++)

    {

        cout<<a[x]<<" ";

    }

}

**QUICK SORT:**

#include <bits/stdc++.h>

using namespace std;

void swap(int\* a, int\* b)

{

    int t = \*a;

    \*a = \*b;

    \*b = t;

}

int partition (int arr[], int low, int high)

{

    int pivot = arr[high];

    int i = (low - 1);

    for (int j = low; j <= high - 1; j++)

    {

        if (arr[j] < pivot)

        {

            i++;

            swap(&arr[i], &arr[j]);

        }

    }

    swap(&arr[i + 1], &arr[high]);

    return (i + 1);

}

void quickSort(int arr[], int low, int high)

{

    if (low < high)

    {

        int pi = partition(arr, low, high);

        quickSort(arr, low, pi - 1);

        quickSort(arr, pi + 1, high);

    }

}

int main()

{

    int n;

    cin >> n;

    int arr[n];

    for (int i =0 ; i < n;i++)

        cin >> arr[i];

    quickSort(arr, 0, n - 1);

    for (int i = 0; i < n; i++)

        cout << arr[i] << " ";

    return 0;

}