**Merge Sort**

**Merge sort Program:**

#include <iostream>

using namespace std;

void Merge(int \*a, int low, int high, int mid)

{

int i, j, k, temp[high-low+1];

i = low;

k = 0;

j = mid + 1;

while (i <= mid && j <= high)

{

if (a[i] < a[j])

{

temp[k] = a[i];

k++;

i++;

}

else

{

temp[k] = a[j];

k++;

j++;

}

}

while (i <= mid)

{

temp[k] = a[i];

k++;

i++;

}

while (j <= high)

{

temp[k] = a[j];

k++;

j++;

}

for (i = low; i <= high; i++)

{

a[i] = temp[i-low];

}

}

void MergeSort(int \*a, int low, int high)

{

int mid;

if (low < high)

{

mid=(low+high)/2;

MergeSort(a, low, mid);

MergeSort(a, mid+1, high);

Merge(a, low, high, mid);

}

}

int main()

{

int n, i;

cout<<"\nEnter the number of element to be sorted: ";

cin>>n;

int arr[n];

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

{

cout<<"Enter element "<<i+1<<": ";

cin>>arr[i];

}

MergeSort(arr, 0, n-1);

cout<<"\nSorted elements are: ";

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

cout<<"->"<<arr[i];

return 0;

}

**Merge short using Recursion:**

#include<stdio.h>

int arr[30];

int merge(int arr[],int l,int m,int h)

{

int arr1[10],arr2[10];

int n1,n2,i,j,k;

n1=m-l+1;

n2=h-m;

for(i=0; i<n1; i++)

arr1[i]=arr[l+i];

for(j=0; j<n2; j++)

arr2[j]=arr[m+j+1];

arr1[i]=9999; // To mark the end of each temporary array

arr2[j]=9999;

i=0;

j=0;

for(k=l; k<=h; k++) {

if(arr1[i]<=arr2[j])

arr[k]=arr1[i++];

else

arr[k]=arr2[j++];

}

return 0;

}

int merge\_sort(int arr[],int low,int high)

{

int mid;

if(low<high) {

mid=(low+high)/2;

merge\_sort(arr,low,mid);

merge\_sort(arr,mid+1,high);

merge(arr,low,mid,high);

}

return 0;

}

int main()

{

int n,i;

printf("Enter the size of array\n");

scanf("%d",&n);

printf("Enter the elements:");

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

scanf("%d",&arr[i]);

merge\_sort(arr,0,n-1);

printf("Sorted array:");

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

printf("%d ",arr[i]);

return 0;

}