Merge Sort

- divide and conquer
- recursion

```
#include<stdio.h>

void merge_sort(int *arr, int low, int high);
void merge(int *arr, int low, int mid, int high);

int main()
{
        int n;
        scanf("%d",&n);
        int a[n];
        for(int i=0 ; i<n ; i++)
        {
                scanf("%d",&a[i]);
        }

        merge_sort(a,0,n-1);

        for(int i=0 ; i<n ; i++)
        {
                printf("%d ",a[i]);
        }
}

void merge_sort(int *arr, int low, int high)
{
   int mid = (high + low) / 2;
   if(low >= high)
   {
       return;
   }
   merge_sort(arr, low, mid);
   merge_sort(arr, mid+1, high);
   merge(arr,low,mid,high);
}

void merge(int *arr, int low, int mid, int high)
{
    int i,j,k,temp[high - low + 1];
    i = low;
    j = mid+1;
    k = 0;

    while((i <= mid) && (j <= high))
    {
        if(arr[i] <= arr[j])
        {
            temp[k] = arr[i];
            i++;
        }
        else
        {
            temp[k] = arr[j];
            j++;
        }
        k++;
    }

    for(/*value of i will be same as of above*/ ; i <= mid ; i++,k++)
    {
        temp[k] = arr[i];
    }
    for(/*value of j will be same as of above*/ ; j <= high ; j++,k++)
    {
        temp[k] = arr[j];
    }
    for(k=0 ; k<=(high-low) ; k++)
    {
        arr[low+k] = temp[k];
    }
}
```