DSA_College\mergeSort.cpp

```cpp
#include<iostream>
using namespace std ;

void merge(int* arr, int l, int mid, int r)
{
    int n1 = mid-l+1 ;
    int n2 = r-mid ;

    int* left = new int[n1] ;
    int* right = new int[n2] ;

    for(int i=0;i<n1;i++)
    {
        left[i] = arr[l+i] ;
    }
    for(int i=0;i<n2;i++)
    {
        right[i] = arr[mid+1+i] ;
    }

    int i=0 ;
    int j=0 ;
    int k=l ;

    while(i<n1 && j<n2)
    {
        if(left[i] < right[j])
        {
            arr[k] = left[i] ;
            i++ ;
            k++ ;
        }
        else{
            arr[k] = right[j] ;
            j++ ;
            k++ ;
        }
    }

    while(i<n1)
    {
        arr[k] = left[i] ;
        i++ ;
        k++ ;
    }

    while(j<n2)
    {
        arr[k] = right[j] ;
        j++ ;
        k++ ;
    }
}

void mergeSort(int arr[], int l, int r)
{
    if(l >= r)
    return ;

    int mid = l+ (r-l)/2 ;
    mergeSort(arr,l,mid) ;
    mergeSort(arr,mid+1,r) ;
    merge(arr,l,mid,r) ;
}

void print(int arr[], int n)
{
    for(int i=0;i<n;i++)
    {
        cout<<arr[i]<<" ";
    }
    cout<<endl;
}

int main()
{
    int arr[] = {24,55,12,9,100,40,32} ;
    int n= sizeof(arr)/sizeof(arr[0]) ;
    mergeSort(arr,0,n-1);
    print(arr,n) ;

    return 0;
}
```