#include<stdio.h>

#include<stdlib.h>

void merge(int arr[],int low,int mid,int high)

{

int temp[50];

int k=low;

int i=low,j=mid+1;

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

{

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

{

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

}else{

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

}

}

while(i<=mid)

{

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

}

while(j<=high )

{

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

}

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

{

arr[low++]=temp[j];

}

return;

}

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

{

if(low<high)

{

int mid=(low+high)/2;

mergeSort(arr,low,mid);

mergeSort(arr,mid+1,high);

merge(arr,low,mid,high);

}

}

int main()

{

int n;

printf("Merge Sort Technique:\n\n");

printf("enter number of elements:\n");

scanf("%d",&n);

int arr[n];

printf("enter %d elements:\n",n);

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

{

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

}

mergeSort(arr,0,n-1);

printf("Sorted array\n\n");

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

{

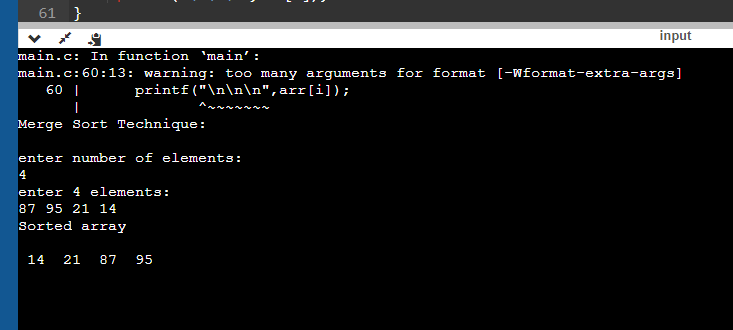
printf(" %d ",arr[i]);

}

printf("\n\n\n",arr[i]);

}

Output:



Quick sort:

#include<stdio.h>

#include<stdlib.h>

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

{

int pivot=arr[high];

int i=(low-1);

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

{

if(arr[j]<=pivot)

{

int temp=arr[++i];

arr[i]=arr[j];

arr[j]=temp;

}

}

int temp=arr[i+1];

arr[i+1]=pivot;

arr[high]=temp;

return i+1;

}

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

{

int pivot=arr[high];

if(low<high)

{

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

quickSort(arr,low,pi-1);

quickSort(arr,pi+1,high);

}

}

int main()

{

int n;

printf("Quick Sort Technique:\n\n");

printf("enter number of elements:\n");

scanf("%d",&n);

int arr[n];

printf("enter %d elements:\n",n);

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

{

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

}

quickSort(arr,0,n-1);

printf("Sorted array\n\n");

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

{

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

}

printf("\n\n\n");

}

Output:

