**SHUBHRADEEP MAITY**

CODE ::

#include<stdio.h>

//-->>for merge sort

void merge(int arr[],int lb,intmid,intub){ int i=lb;

int j=mid+1; int k=lb;

int temp[100]; while(i<=mid && j<=ub){ if(arr[i]<=arr[j]){

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

}

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

} k++;

}

while(j<=ub){temp[k]=arr[j]; j++;k++;

}

while(i<=mid){temp[k]=arr[i]; i++;k++;

}

for(int i=0;i<=ub;i++){ arr[i]=temp[i];

}

}

void mergeSort(int arr[],int lb,intub){ int mid;

if(lb<ub){

mid=(lb+ub)/2; mergeSort(arr,lb,mid); mergeSort(arr,mid+1,ub); merge(arr,lb,mid,ub);

}

}

//--->> for quick sort

int partition(int arr[],int lb,intub){ int pivot=arr[lb];

int start = lb; int end=ub;

while(start<end){ while(arr[start]<=pivot){ start++;

}

while(arr[end]>pivot){

end--;

}

if(start<end){

int temp=arr[end]; arr[end]=arr[start]; arr[start]=temp;

}

}

int temp2=arr[end]; arr[end]=arr[lb]; arr[lb]=temp2; return end;

}

void quick\_sort(int arr[],int lb,intub){ if(lb<ub){

int loc= partition(arr,lb,ub); quick\_sort(arr,lb,loc-1); quick\_sort(arr,loc+1,ub);

}

}

int main(){ int n;

printf("Enter the size of the array :: "); scanf("%d",&n);

int arr[n];

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

printf("\nEnter the array element %d : ",i); scanf("%d",&arr[i]);

}

printf("\Enter 1 for quick sort\nEnter 2 for merge sort\nEnter your choice :: "); int ch;

scanf("%d",&ch); switch(ch){

case 1: quick\_sort(arr,0,n-1); break;

case 2: mergeSort(arr,0,n-1);

}

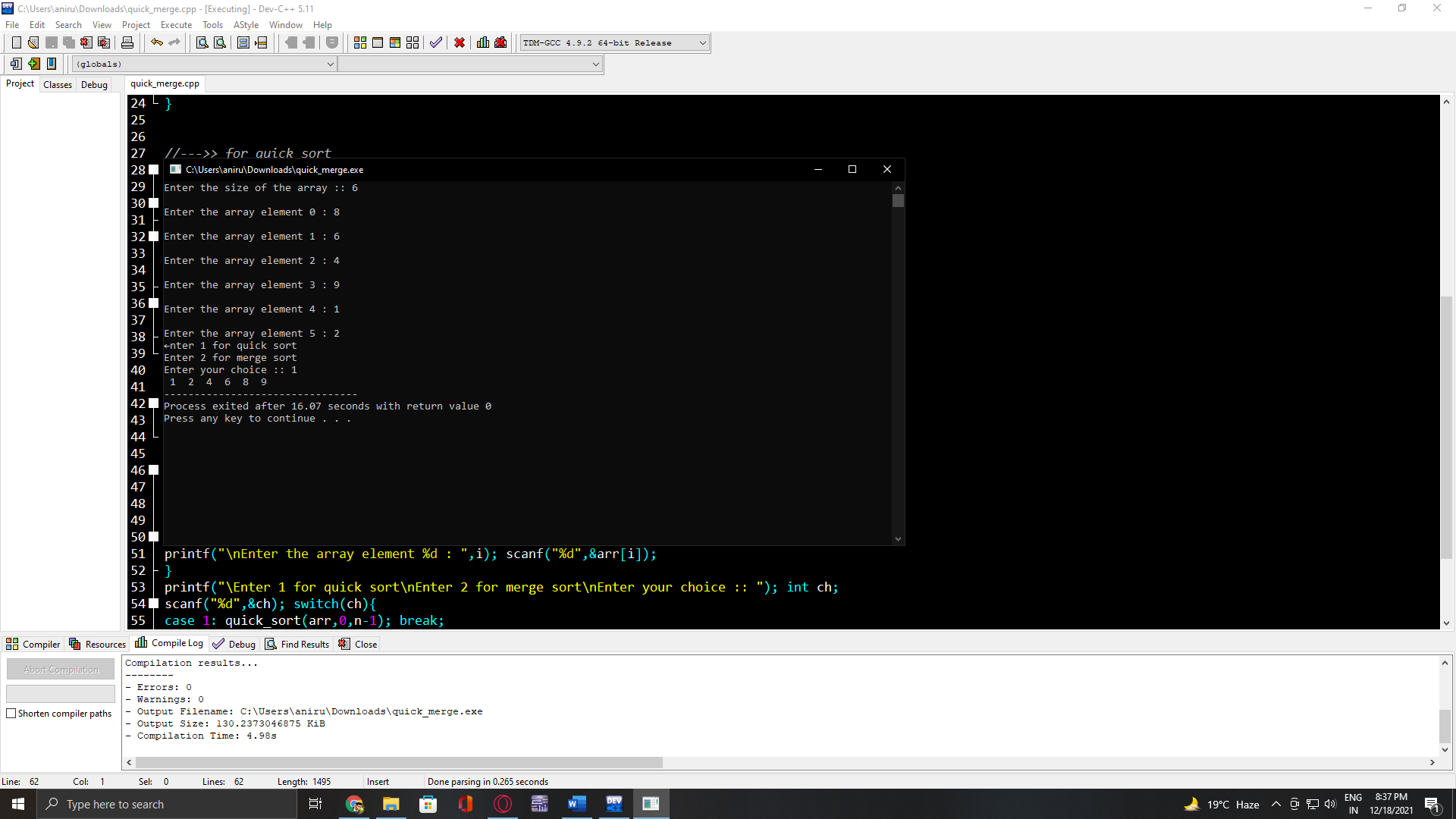
for(int i=0;i<n;i++){ printf(" %d ",arr[i]);

}

return 0;

}

OUTPUT ::

****

