**Programs**

1. Write a program to implement quick sort.

Ans:

#include<stdio.h>

#include<conio.h>

void quick(int a[10],int first,int last)

{

int i,j,pivot,temp;

if(first<last)

{

pivot=first;

i=first;

j=last;

while(i<j)

{

while(a[i]<=a[pivot]&&i<last)

i++;

while(a[j]>a[pivot])

j--;

if(i<j)

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

}

}

temp=a[pivot];

a[pivot]=a[j];

a[j]=temp;

quick(a,first,j-1);

quick(a,j+1,last);

}

}

void display(int a[],int n)

{

int i;

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

printf("%d\t",a[i]);

}

void main(){

int i,n,a[10];

clrscr();

printf("Enter how many elements you want to insert:");

scanf("%d",&n);

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

{

printf("Enter element:");

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

}

printf("Before sorting:\n");

display(a,n);

quick(a,0,n-1);

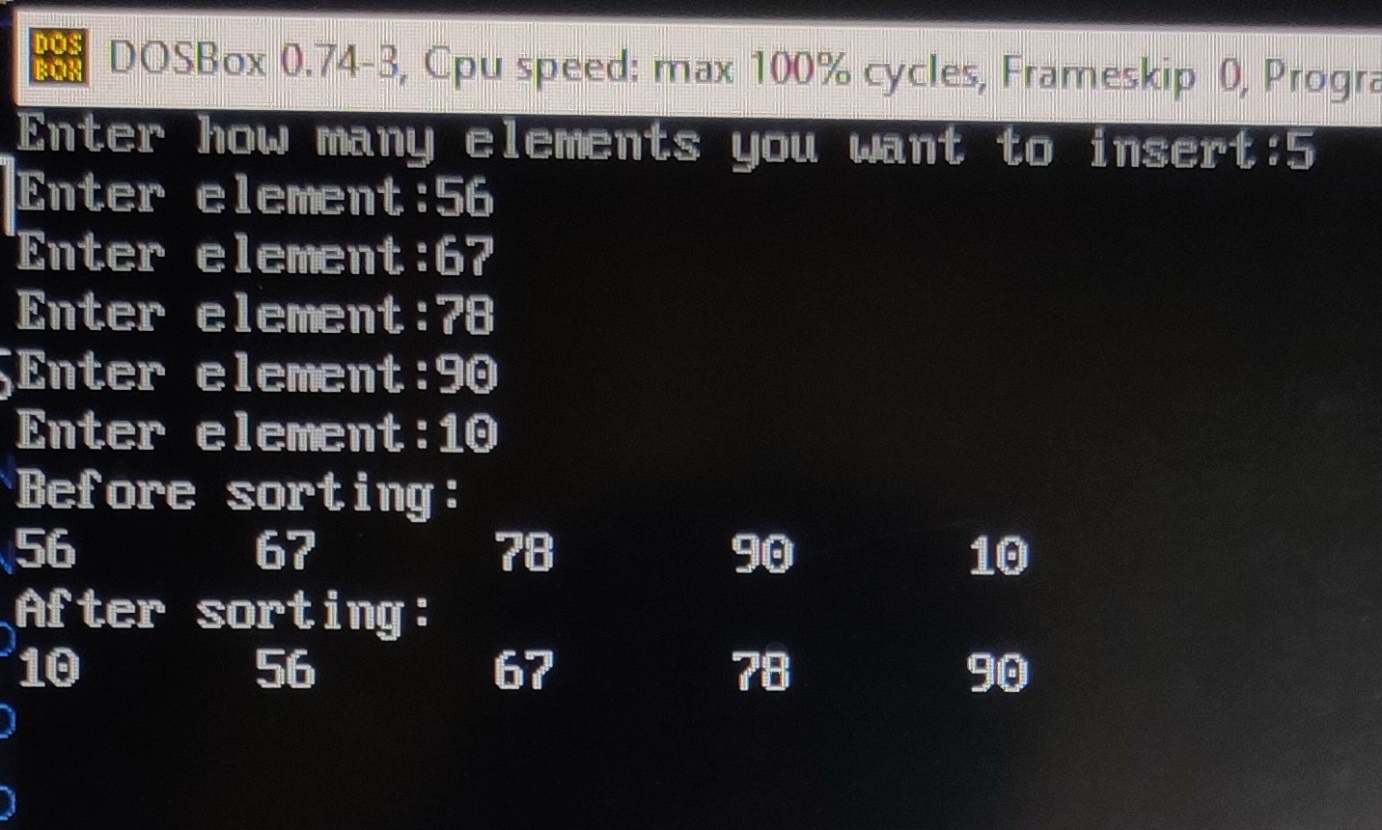
printf("\nAfter sorting:\n");

display(a,n);

getch();

}

Output:



1. Write a program to implement merge sort.

Ans:

#include <stdio.h>

#include<conio.h>

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

{

int i,j,k,b[20];

i=low;

j=mid + 1;

k=low;

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

{

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

{

b[k]=a[i];

i++;

k++;

}

else

{

b[k]=a[j];

j++;

k++;

}

}

while(i<=mid)

{

b[k]=a[i];

k++;

i++;

}

while(j<=high)

{

b[k]=a[j];

k++;

j++;

}

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

a[i]=b[i];

}

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,mid,low,high);

}

}

void display(int a[],int n)

{

int i;

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

printf("%d\t",a[i]);

}

void main()

{

int a[10],n,i;

clrscr();

printf("Enter how many elements you want to insert:");

scanf("%d",&n);

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

{

printf("Enter element:");

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

}

printf("Before sorting:\n");

display(a,n);

mergeSort(a,0,n-1);

printf("\nBefore sorting:\n");

display(a, n);

getch();

}

Output:

