MERGE SORT

```c
#include<stdio.h>

#include<conio.h>

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

{

   int i,j,k,c[100];

   i=low;

   j=mid+1;

   k=low;

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

   {

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

       c[k++]=a[i++];

     else{

       c[k++]=a[j++];

     }

   }

   while(i<=mid)

     c[k++]=a[i++];
```

```c
    while(j<=high)
        c[k++]=a[j++];
    for(i=low;i<=high;i++)
    {
        a[i]=c[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,low,mid,high);
    }
}
```

```c
void main()
{
    int a[100],n,i;
    printf("Enter the no.of elements to be sorted:\n");
    scanf("%d",&n);
    printf("Enter the elements:");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    mergesort(a,0,n-1);
    printf("Sorted array is:");
    for(i=0;i<n;i++)
    {
        printf("%d\t",a[i]);
    }
    getch();
}
```

QUICK SORT

```c
#include<stdio.h>

#include<conio.h>

void qsort(int a[],int low,int high)

{

    int mid;

    if(low<high)

    {

        mid=partition(a,low,high);

        qsort(a,low,mid-1);

        qsort(a,mid+1,high);

    }

}
```

```c
int partition(int a[],int low,int high)
{
    int i,j,temp,pivot;
    pivot=a[low];
    i=low+1;
    j=high;
    while(i<=j)
    {
        while(a[i]<=pivot)
        {
            i++;
        }
        while(a[j]>pivot)
            j--;
        if(i<j)
        {
            temp=a[i];
            a[i]=a[j];
            a[j]=temp;
        }
```
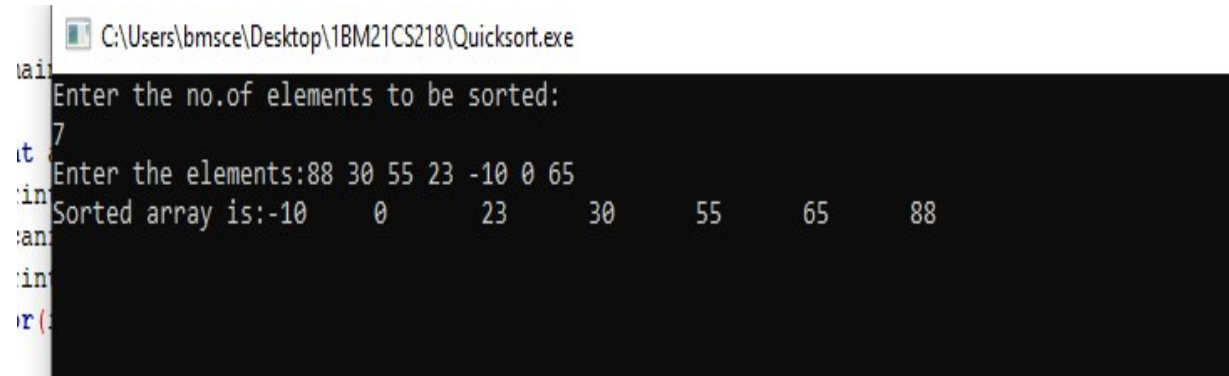
```c
    }
    temp=a[low];
    a[low]=a[j];
    a[j]=temp;
    return j;
}
void main()
{
    int a[100],n,i;
    printf("Enter the no.of elements to be sorted:\n");
    scanf("%d",&n);
    printf("Enter the elements:");
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    qsort(a,0,n-1);
    printf("Sorted array is:");
    for(i=0;i<n;i++)
    {
```

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

    }

    getch();

}
```