```
/*Program4:-Write a program for implementing the following sorting methods
to arrange a list of integers in ascending order:-
1.Selection Sort
2.Insertion Sort
3.Quick Sort
4.Merge Sort*/
#include<stdio.h>
#include<conio.h>
void Selection_Sort(int [],int);
void Insertion_Sort(int [],int);
void Quick_Sort(int [],int ,int);
void Merge_Sort(int a[],int n);
void main()
{
 int a[50],i,n,choice,lb,ub;
 char ch;
 clrscr();
 do
 {
 printf("\nEnter total number of elements to be sorted\n");
 scanf("%d",&n);
 printf("Enter elements to be sorted\n");
 for(i=0;i<n;i++)
  scanf("%d",&a[i]);
  printf("\nEnter your choice for sorting algorihtm\n1.Selection Sort \n2.Insertion Sort \n3.Quick Sort \n4.M
erge Sort\n");
  fflush(stdin);
  scanf("%d",&choice);
  switch(choice)
  {
   case 1:
    Selection_Sort(a,n);
    break;
   case 2:
    Insertion_Sort(a,n);
    break;
   case 3:
    lb=0;
    ub=n-1;
    printf("\nArray of elements before sorting is:\n");
    for(i=0;i<n;i++)
      printf("%d\t",a[i]);
    Quick_Sort(a,lb,ub);
    printf("\nArray of elememts after sorting is:\n");
    for(i=0;i<n;i++)
      printf("%d\t",a[i]);
    break;
   case 4:
    Merge_Sort(a,n);
    break;
   default:
    printf("Entered Wrong choice\n");
  }
  printf("\nDo you want to continue, press y/n\n");
  fflush(stdin);
```

```
  scanf("%c",&ch);
 }while(ch=='Y'||ch=='y');
 getch();
}
void Selection_Sort(int a[],int n)
{
 int i,j,index,large;
 printf("\nArray of elememts before sorting is:\n");
 for(i=0;i<n;i++)
  printf("%d\t",a[i]);
 for(i=n-1;i>0;i--)
 {
  index=0;
  large=a[0];
  for(j=1;j<=i;j++)
  {
   if(a[j]>large)
   {
    index=j;
    large=a[j];
   }
  }
 a[index]=a[i];
 a[i]=large;
 }
 printf("\nArray of elememts after sorting is:\n");
 for(i=0;i<n;i++)
  printf("%d\t",a[i]);
}
void Insertion_Sort(int a[],int n)
{
  int i,j,temp;
  printf("\nArray of elememts before sorting is:\n");
  for(i=0;i<n;i++)
   printf("%d\t",a[i]);
  for(i=1;i<n;i++)
  {
   if(a[i]<a[i-1])
     {
      j=i;
      temp=a[j];
      do
      {
       a[j]=a[j-1];
       --j;
      }while(j>0 && a[j-1]>temp);
     a[j]=temp;
    }
  }
  printf("\nArray of elememts after sorting is:\n");
  for(i=0;i<n;i++)
   printf("%d\t",a[i]);
}
void Quick_Sort(int a[],int lb,int ub)
{
```

```
 int L,R,temp,X,i;
 if(lb>=ub)
  return;
 L=lb;
 R=ub;
 X=a[lb];
 while(L<R)
 {
  while(a[L]<=X && L<R)
   L++;
  while(a[R]>X)
   --R;
  if(L<R)
   {
    temp=a[L];
    a[L]=a[R];
    a[R]=temp;
   }
  }
 a[lb]=a[R];
 a[R]=X;
 Quick_Sort(a,lb,R-1);
 Quick_Sort(a,R+1,ub);
}
void Merge_Sort(int a[],int n)
{
 int i,j,L1,L2,u1,u2,K,temp[25],size=1;
 printf("\nArray of elememts before sorting is:\n");
 for(i=0;i<n;i++)
  printf("%d\t",a[i]);
 while(size<n)
 {
  L1=K=0;
  while(L1+size<n)
   {
   L2=L1+size;
   u1=L2-1;
   if(L2+size-1<n)
    u2=L2+size-1;
   else
    u2=n-1;
   for(i=L1,j=L2;i<=u1 && j<=u2;K++)
    {
    if(a[i]<a[j])
     temp[K]=a[i++];
    else
     temp[K]=a[j++];
    }
   for(;i<=u1;K++)
    temp[K]=a[i++];
   for(;j<=u2;K++)
    temp[K]=a[j++];
   L1=u2+1;
  }
 for(i=L1;i<n;K++)
```

```
  temp[K]=a[i++];
 for(i=0;i<n;i++)
  a[i]=temp[i];
  size=size*2;
 }
 printf("\nArray of elememts after sorting is:\n");
 for(i=0;i<n;i++)
 printf("%d\t",a[i]);
}
```

```
Enter total number of elements to be sorted
4
Enter elements to be sorted
90
80
70
10

Enter your choice for sorting algorihtm
1.Selection Sort
2.Insertion Sort
3.Quick Sort
4.Merge Sort
1

Array of elememts before sorting is:
90      80      70      10
Array of elememts after sorting is:
10      70      80      90
Do you want to continue, press y/n
Y_
```

```
Enter total number of elements to be sorted
6
Enter elements to be sorted
90
30
40
10
20
60

Enter your choice for sorting algorihtm
1.Selection Sort
2.Insertion Sort
3.Quick Sort
4.Merge Sort
2

Array of elememts before sorting is:
90      30      40      10      20      60
Array of elememts after sorting is:
10      20      30      40      60      90
Do you want to continue, press y/n
y_
```

```
Enter total number of elements to be sorted
7
Enter elements to be sorted
90
80
70
10
20
100
200

Enter your choice for sorting algorihtm
1.Selection Sort
2.Insertion Sort
3.Quick Sort
4.Merge Sort
3

Array of elements before sorting is:
90      80      70      10      20      100     200
Array of elememts after sorting is:
10      20      70      80      90      100     200
Do you want to continue, press y/n
y_
```

```
Enter total number of elements to be sorted
8
Enter elements to be sorted
90
80
10
20
30
70
60
50

Enter your choice for sorting algorihtm
1.Selection Sort
2.Insertion Sort
3.Quick Sort
4.Merge Sort
4

Array of elememts before sorting is:
90      80      10      20      30      70      60      50
Array of elememts after sorting is:
10      20      30      50      60      70      80      90
Do you want to continue, press y/n
N
```