```
#include<stdio.h>
#include<stdlib.h>
int a[20],n;

void insertionsort()
{
int i,j,temp;
for(i=1;i<n;i++)
{
temp=a[i];
j=i-1;
while(temp<a[j]&&(j>=0))
{
a[j+1]=a[j];
j=j-1;
}
a[j+1]=temp;
}
printf("The Sorted Array using insertion sort is:\n");
for(i=0;i<n;i++)
printf("%d\t",a[i]);
}


void selectionsort()
{
int i,j,temp;
for(i=0;i<n;i++)
{
for(j=i+1;j<n;j++)
{
if(a[i]>a[j])
{
temp = a[i];
a[i]=a[j];
a[j]=temp;
} } }
printf("The Sorted Array using selection sort is:\n");
for(i=0;i<n;i++)
{
printf("%d\t",a[i]);
}
}


void quicksort(int a[25],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;
 quicksort(a,first,j-1);
 quicksort(a,j+1,last);
 }
}
int mquicksort()
{
 int i;
 quicksort(a,0,n-1);
 printf("The Sorted Array using quick sort is:\n");
 for(i=0;i<n;i++)
 printf("%d\t",a[i]);
 return 0;
}


void bubblesort()
 {
   int i, j, temp;
   for(i = 0; i < n; i++)
    {
      for(j = i+1; j < n; j++)
       {
          if(a[j] < a[i])
          {
             temp = a[i];
             a[i] = a[j];
             a[j] = temp;
          }
       }
    }
   printf("The Sorted array using bubble sort is:\n");
   for(i=0;i<n;i++)
   printf("%d\t",a[i]);

 }


void main()
 {
              int c,i;
              printf("\nEnter the size of array:");
              scanf("%d",&n);
              printf("\nEnter the elements of array:");
              for(i=0;i<n;i++)
              scanf("%d",&a[i]);
```

```
        while(1)
         {
                printf("\n1.InsertionSort\n2.SelectionSort\n3.QuickSort\n4.BubbleSort\n5.Exit");

                printf("\nEnter a choice:");
                scanf("%d",&c);
                switch(c)
                {
                        case 1:insertionsort();
                        break;
                        case 2:selectionsort();
                        break;
                        case 3:mquicksort();
                        break;
                        case 4:bubblesort();
                        break;
                        case 5:exit(0);
                        default:printf("Invalid choice");

                }


         }
}
```