Bubble sort:

```
#include<iostream.h>
#include<conio.h>
class BSort
{
        int arr[10];
        public:
                void getdata();
                void display();
                void banketsort();
                void bubblesort();
                void search1();
                void search2(int);
                int search3();
                int search4(int);
} ;
void BSort::getdata()
{
        int i;
        cout<<"\nEnter 10 elements in array\n";
        for(i=0;i<10;i++)
                cin>>arr[i];
}
```

```
void BSort::display()
{
	int i;
	cout<<"\nElements in array: \n";
	for(i=0;i<10;i++)
		cout<<arr[i]<<"\t";
}

void BSort::banketsort()
{
	int i,j,temp;
	for(i=0;i<10;i++)
	{
		for(j=i+1;j<10;j++)
		{
			if(arr[i]>arr[j])
			{
				temp=arr[i];
				arr[i]=arr[j];
				arr[j]=temp;
			}
		}
	}
}
```

```
void BSort::bubblesort()
{
	int i,j,temp;
	for(i=0;i<10;i++)
	{
		for(j=0;j<9;j++)
		{
			if(arr[j]>arr[j+1])
			{
				temp=arr[j];
				arr[j]=arr[j+1];
				arr[j+1]=temp;
			}
		}
	}
}

void BSort::search1()
{
	int num,pos=1,flag=0;
	cout<<"\nEnter element to search: ";
	cin>>num;
	for(int i=0;i<10;i++)
```

```
	{
		if(arr[i]==num)
		{
			flag=1;
			break;
		}
		pos++;
	}
	if(flag==1)
	{
		cout<<"\nElement found at pos: "<<pos;
	}
	else
	{
		cout<<"\nElement not found";
	}
}

void BSort::search2(int num)
{
	int flag=0,pos=1;
	for(int i=0;i<10;i++)
	{
		if(arr[i]==num)
```

```
            {
                flag=1;
                break;
            }
            pos++;
        }
        if(flag==1)
        {
            cout<<"\nElement found at pos: "<<pos;
        }
        else
        {
            cout<<"\nElement not found";
        }
}

int BSort::search3()
{
        int num,pos=1,flag=0;
        cout<<"\nEnter element to search: ";
        cin>>num;
        for(int i=0;i<10;i++)
        {
            if(arr[i]==num)
```

```
            {
                flag=1;
                break;
            }
            pos++;
        }
        if(flag==1)
        {
            return pos;
        }
        else
        {
            return -1;
        }
}

int BSort::search4(int num)
{
        int flag=0,pos=1;
        for(int i=0;i<10;i++)
        {
            if(arr[i]==num)
            {
                flag=1;
```

```
                    break;
                }
                pos++;
        }
        if(flag==1)
        {
                return pos;
        }
        else
        {
                return -1;
        }
}

void main()
{
        clrscr();
        int opt,num,result;
        BSort b;
        b.getdata();
        cout<<"\nElements without sorting:\n";
        b.display();
        cout<<"\n1.Banket Sort\n2.Bubble
Sort\n3.Search1\n4.search2\n5.Search3\n6.search4\n";
        cout<<"\nEnter the Option to perform the operation: ";
```

```
cin>>opt;
switch(opt)
{
        case 1:
                        b.banketsort();
                        cout<<"\nElements after sorting:\n";
                        b.display();
                        break;
        case 2:
                        b.bubblesort();
                        cout<<"\nElements after sorting:\n";
                        b.display();
                        break;
        case 3:
                        b.search1();
                        break;
        case 4:
                        cout<<"\nEnter the element to search: ";
                        cin>>num;
                        b.search2(num);
                        break;
        case 5:
                        result=b.search3();
                        if(result==-1)
```

```
                {
                        cout<<"\nElement not found";
                }
                else
                {
                        cout<<"\nElement found at pos: "<<result;
                }
                break;
        case 6:
                cout<<"\nEnter the element to search: ";
                cin>>num;
                result=b.search4(num);
                if(result==-1)
                {
                        cout<<"\nElement not found";
                }
                else
                {
                        cout<<"\nElement found at pos: "<<result;
                }
                break;
    }
    getch();
}
```

SHELL SORT :-

```
#include<iostream.h>
#include<conio.h>
class shellsort
{
private:
 int arr[10];
public:
 void getdata();
 void display();
 void shell(void);
};
void shellsort::getdata()
{
 int i;
 cout<<"Enter elements in array\n"<<endl;
 for(i=0;i<=9;i++)
 {
 cin>>arr[i];
 }
}
void shellsort::display()
{
 int i;
```

```
 cout<<"Elements of the array\n"<<endl;
 for(i=0;i<=9;i++)
 {
 cout<<arr[i]<<endl;
 }
}
void shellsort::shell(void)
{
 int i,j,k,temp,gap,n=10;
 gap=n/2;
 while(gap!=1)
 {
 for(i=0;i<=9-gap;i++)
 {
 if(arr[i]>arr[i+gap])
 {
 temp=arr[i];
 arr[i]=arr[i+gap];
 arr[i+gap]=temp;
 }
 }
gap=gap/2;
 }
 for(k=0;k<=4;k++)
```

```
 {
 for(i=0;i<=8;i++)
 {
 if(arr[i]>arr[i+1])
 {
 temp=arr[i];
 arr[i]=arr[i+1];
 arr[i+1]=temp;
 }
 }
 }
 }
void main(void)
{
clrscr();
shellsort s;
s.getdata();
s.display();
s.shell();
s.display();
getch();
}
```

RADIX SORT :-

```
#include<iostream.h>
#include<conio.h>
class rsort
{
private:
 int arr[5];
public:
 void getdata();
 void display();
 void radix(void);
};
void rsort::getdata()
{
 int i;
 cout<<"Enter elements in array\n"<<endl;
 for(i=0;i<5;i++)
 {
 cin>>arr[i];
 }
}
void rsort::display()
{
 int i;
```

```
 cout<<"Elements of the array\n"<<endl;
 for(i=0;i<=4;i++)
 {
 cout<<arr[i]<<endl;
 }
}
void rsort::radix(void)
{
 int i,j,r=1,temp;
 while(r!=1000)
 {
 for(i=0;i<=4;i++)
 {
 for(j=i+1;j<=4;j++)
 {
 if(((arr[i]/r)%10)>((arr[j]/r)%10))
 {
 temp=arr[i];
 arr[i]=arr[j];
 arr[j]=temp;
 }
 }
}
r=r*10;
```

```
}
}
void main()
{
clrscr();
rsort s;
s.getdata();
s.display();
s.radix();
cout<<"\n Elements after sorting";
s.display();
getch();
}
```

Selection Sort :

```
#include<iostream.h>
#include<conio.h>
class SelectionSort
{
int arr[5];
public:
void getdata();
void display();
```

```
void selectionsort();
};
void SelectionSort::selectionsort()
{
int i,min,j,count=5,temp;
for(i=0;i<5;i++)
{
min=i;
for(j=i+1;j<5;j++)
{
if(arr[j]<arr[min])
min=j;
}
temp=arr[i];
arr[i]=arr[min];
arr[min]=temp;
}
}
void SelectionSort::getdata()
{
cout<<"\nEnter 5 elements in array: \n";
for(int i=0;i<5;i++)
cin>>arr[i];
}
```

```
void SelectionSort::display()
{
cout<<"\nElements in array: \n";
for(int i=0;i<5;i++)
cout<<arr[i]<<"\t";
}
void main()
{
clrscr();
SelectionSort s;
s.getdata();
s.selectionsort();
s.display();
getch();
}
```

Quick Sort :

```
#include<iostream.h>
#include<conio.h>
class QSORT
{
public:
void quick(int[],int,int);
void display(int[],int);
};
void QSORT::quick(int a[], int start, int end)
{
 if (start < end)
 {
int pivot = a[end]; // pivot element
 int i = (start - 1);
 for (int j = start; j <= end - 1; j++)
 {
// If current element is smaller than the pivot
if (a[j] < pivot)
{
 i++; // increment index of smaller element
 int t = a[i];
 a[i] = a[j];
 a[j] = t;
```

```
}
 }
 int t = a[i+1];
 a[i+1] = a[end];
 a[end] = t;
 int p=i + 1;
quick(a, start, p - 1);
quick(a, p + 1, end);
 }
}
void QSORT::display(int a[], int n)
{
 int i;
 for (i = 0; i < n; i++)
cout<<a[i]<< " ";
}
void main()
{
clrscr();
QSORT q;
 int a[] = { 23, 8, 28, 13, 18, 26 };
 int n = sizeof(a) / sizeof(a[0]);
 cout<<"Before sorting array elements are - \n";
 q.display(a, n);
```

```
 q.quick(a, 0, n - 1);
 cout<<"\nAfter sorting array elements are - \n";
 q.display(a, n);
 getch();
}
```

Insertion Sort :

```
#include<iostream.h>
#include<conio.h>
// insertion sort
void insertionSort(int arr[], int n)
{
        int i, key, j;
        for (i = 1; i < n; i++)
        {
                key = arr[i];
                j = i - 1;
                while (j >= 0 && arr[j] > key)
                {
                        arr[j + 1] = arr[j];
                        j = j - 1;
                }
                arr[j + 1] = key;
```

```
        }
}
void printArray(int arr[], int n)
{
        int i;
        for (i = 0; i < n; i++)
                cout << arr[i] << " ";
        cout << endl;
}

void main()
{
        clrscr();
        int arr[] = { 12, 11, 13, 5, 6 };
        int N = sizeof(arr) / sizeof(arr[0]);

        insertionSort(arr, N);
        printArray(arr, N);

        getch();
}
```