Merge Sort

```
#include <iostream.h>
int a[50];
void merge(int,int,int);
void merge_sort(int low,int high)
{
   int mid;
   if(low<high)
   {
      mid=(low+high)/2;
      merge_sort(low,mid);
      merge_sort(mid+1,high);
      merge(low,mid,high);
   }
}
void merge(int low,int mid,int high)
{
   int h,i,j,b[50],k;
   h=low;
   i=low;
   j=mid+1;
   while((h<=mid)&&(j<=high))
   {
      if(a[h]<=a[j])
      {
      b[i]=a[h];
      h++;
      }
      else
      {
      b[i]=a[j];
      j++;
      }
      i++;
   }
   if(h>mid)
   {
      for(k=j;k<=high;k++)
      {
      b[i]=a[k];
      i++;
      }
   }
   else
   {
      for(k=h;k<=mid;k++)
      {
      b[i]=a[k];
      i++;
      }
   }
   for(k=low;k<=high;k++)
   a[k]=b[k];
}
void main()
{
   int num,i;

   cout<<"Please Enter THE NUMBER OF ELEMENTS : ";
   cin>>num;
   cout<<endl;
   cout<<"Enter ELEMENTS : \n";
   for(i=1;i<=num;i++)
   {
      cin>>a[i] ;
```

Merge Sort

```
        {
        b[i]=a[j];
        j++;
        }
        i++;
    }
    if(h>mid)
    {
        for(k=j;k<=high;k++)
        {
        b[i]=a[k];
        i++;
        }
    }
    else
    {
        for(k=h;k<=mid;k++)
        {
        b[i]=a[k];
        i++;
        }
    }
    for(k=low;k<=high;k++)
    a[k]=b[k];
}
void main()
{
    int num,i;

    cout<<"Please Enter THE NUMBER OF ELEMENTS : ";
    cin>>num;
    cout<<endl;
    cout<<"Enter ELEMENTS : \n";
    for(i=1;i<=num;i++)
    {
        cin>>a[i] ;
    }
    merge_sort(1,num);
    cout<<endl;
    cout<<"Sorted list  :"<<endl;
    cout<<endl<<endl;
    for(i=1;i<=num;i++)
    cout<<a[i]<<"   ";
    cout<<endl<<endl<<endl<<endl;
}
```

OUTPUT :

Please Enter THE NUMBER OF ELEMENTS : 5

Enter ELEMENTS :
1
4
2
6
8

Sorted list :

1 2 4 6 8

Quick Sort

```
#include <iostream.h>
#include <conio.h>

#define MAXSIZE 500


 int elements[MAXSIZE];

 void quickSort(int elements[], int left, int right)
 {
   int pivot, l, r;
   l = left;
   r = right;
   pivot = elements[left];
   while (left < right)
    {
     while ((elements[right] >= pivot) && (left < right)
)
     right--;
     if (left != right)
      {
       elements[left] = elements[right];
       left++;
      }
     while ((elements[left] <= pivot) && (left < right))
     left++;
     if(left != right)
      {
        elements[right] = elements[left];
        right--;
      }
    }
   elements[left] = pivot;
   pivot = left;
   left = l;
   right = r;
   if (left < pivot)
   quickSort(elements, left, pivot - 1);
   if (right > pivot)
   quickSort(elements, pivot + 1, right);
}


 int main()
 {
   int i, maxsize;
   cout<<"\nHow many elements you want to sort: ";
   cin>>maxsize;
   cout<<"\nEnter the values one by one: ";

   for (i = 0; i < maxsize; i++)
    {
     cout<<"\nEnter element "<<i<<" : ";
     cin>>elements[i];
    }
   cout<<"\n Array before sorting:\n";
   for (i = 0; i < maxsize; i++)
   cout<<"["<<elements[i]<<"]\n";

   quickSort(elements,0, maxsize-1);
   cout<<"\n Array after sorting:\n";
   for (i = 0; i < maxsize; i++)
   cout<<"["<<elements[i]<<"]\n";
```

Quick Sort

```
        elements[right] = elements[left];
        right--;
      }
    }
    elements[left] = pivot;
    pivot = left;
    left = l;
    right = r;
    if (left < pivot)
    quickSort(elements, left, pivot - 1);
    if (right > pivot)
    quickSort(elements, pivot + 1, right);
}


 int main()
 {
   int i, maxsize;
   cout<<"\nHow many elements you want to sort: ";
   cin>>maxsize;
   cout<<"\nEnter the values one by one: ";

   for (i = 0; i < maxsize; i++)
    {
      cout<<"\nEnter element "<<i<<" : ";
      cin>>elements[i];
    }
   cout<<"\n Array before sorting:\n";
   for (i = 0; i < maxsize; i++)
   cout<<"["<<elements[i]<<"]\n";

   quickSort(elements,0, maxsize-1);
   cout<<"\n Array after sorting:\n";
   for (i = 0; i < maxsize; i++)
   cout<<"["<<elements[i]<<"]\n";
 return 0;
 }
```

OUTPUT :

```
How many elements you want to sort: 4

Enter the values one by one:

Enter element 0 : 12

Enter element 1 : 22

Enter element 2 : 11

Enter element 3 : 56

 Array before sorting:
[12]
[22]
[11]
[56]

 Array after sorting:
[11]
[12]
[22]
[56]
```

Bucket or Radix Sort

```
#include <iostream.h>

#define MAX 100

  void print(int *a, int n)
  {
     int i;
     for (i = 0; i < n; i++)
     cout<<a[i]<<"\t :";
  }
  void radix_sort(int *a, int n)
  {
     int i, b[MAX], m = 0, exp = 1;
     for (i = 0; i < n; i++)
     {
        if (a[i] > m)
        m = a[i];
     }
     while (m / exp > 0)
     {
        int box[10] = { 0 };
        for (i = 0; i < n; i++)
        box[a[i] / exp % 10]++;
        for (i = 1; i < 10; i++)
        box[i] += box[i - 1];
        for (i = n - 1; i >= 0; i--)
        b[--box[a[i] / exp % 10]] = a[i];
        for (i = 0; i < n; i++)
        a[i] = b[i];
        exp *= 10;

     }
  }
  int main()
  {
     int arr[MAX];
     int i, num;
     cout<<"\nEnter total elements (num < "<<MAX<<") : ";
       cin>>num;

     cout<<"\n Enter Elements : ";
     for (i = 0; i < num; i++)
     cin>>arr[i];
     cout<<"\n ARRAY : ";
     print(&arr[0], num);
     radix_sort(&arr[0], num);
     cout<<"\n\n SORTED : ";
     print(&arr[0], num);
     return 0;
  }
```

OUTPUT :

```
Enter How many Numbers : 4

Enter elements to be sorted:
1
5
3
6

The array of elements before sorting :
1 5 3 6
```