**1.QUICK SORT:**

#include<stdio.h>

#include<conio.h>

void quicksort(int number[25],int first,int last)

{

int i, j, pivot, temp;

if(first<last)

{

pivot=first;

i=first;

j=last;

while(i<j)

{

while(number[i]<=number[pivot]&&i<last)

i++;

while(number[j]>number[pivot])

j--;

if(i<j)

{

temp=number[i];

number[i]=number[j];

number[j]=temp;

}

}

temp=number[pivot];

number[pivot]=number[j];

number[j]=temp;

quicksort(number,first,j-1);

quicksort(number,j+1,last);

}

}

int main()

{

int i, count, number[25];

printf("Enter some elements: ");

scanf("%d",&count);

printf("Enter %d elements: ", count);

for(i=0;i<count;i++)

scanf("%d",&number[i]);

quicksort(number,0,count-1);

printf("The Sorted Order is: ");

for(i=0;i<count;i++)

printf(" %d",number[i]);

return 0;

}

**OUTPUT:**

Enter some elements: 10

Enter 10 elements: 56 1 23 89 9 3 87 45 8 34

The Sorted Order is:  1 3 8 9 23 34 45 56 87 89

**2.MERGE SORT:**

#include <stdio.h>

#include<conio.h>

void mergeSort(int [], int, int, int);

void partition(int [],int, int);

int main()

{

int list[50];

int i, size;

printf("Enter total number of elements:");

scanf("%d", &size);

printf("Enter the elements:\n");

for(i = 0; i < size; i++)

{

scanf("%d", &list[i]);

}

partition(list, 0, size - 1);

printf("After merge sort:\n");

for(i = 0;i < size; i++)

{

printf("%d ",list[i]);

}

return 0;

}

void partition(int list[],int low,int high)

{

int mid;

if(low < high)

{

mid = (low + high) / 2;

partition(list, low, mid);

partition(list, mid + 1, high);

mergeSort(list, low, mid, high);

}

}

void mergeSort(int list[],int low,int mid,int high)

{

int i, mi, k, lo, temp[50];

lo = low;

i = low;

mi = mid + 1;

while ((lo <= mid) && (mi <= high))

{

if (list[lo] <= list[mi])

{

temp[i] = list[lo];

lo++;

}

else

{

temp[i] = list[mi];

mi++;

}

i++;

}

if (lo > mid)

{

for (k = mi; k <= high; k++)

{

temp[i] = list[k];

i++;

}

}

else

{

for (k = lo; k <= mid; k++)

{

temp[i] = list[k];

i++;

}

}

for (k = low; k <= high; k++)

{

list[k] = temp[k];

}

}

**OUTPUT:**

Enter total number of elements:10

Enter the elements:

32 12 45 54 73 37 82 28 91 19

After merge sort:

12   19   28   32   37   45   54   73   82   91