Expt no: 1

**PROGRAM: C program to implement Merge Sort.**

**Input:**

#include <stdio.h>

#define max 10

int a[10];

int b[10];

void merging(int low, int mid, int high)

{

int l1, l2, i;

for(l1 = low, l2 = mid + 1, i = low; l1 <= mid && l2 <= high; i++)

{

if(a[l1] <= a[l2])

b[i] = a[l1++];

else

b[i] = a[l2++];

}

while(l1 <= mid)

b[i++] = a[l1++];

while(l2 <= high)

b[i++] = a[l2++];

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

a[i] = b[i];

}

void sort(int low, int high)

{

int mid;

if(low < high)

{

mid = (low + high) / 2;

sort(low, mid);

sort(mid+1, high);

merging(low, mid, high);

}

else

{

return;

}

}

int main()

{

int i,n;

printf("Enter the no of elements: \n");

scanf("%d",&n);

printf("Enter the elements in an array: \n");

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

{

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

}

printf("List before sorting\n");

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

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

sort(0, max);

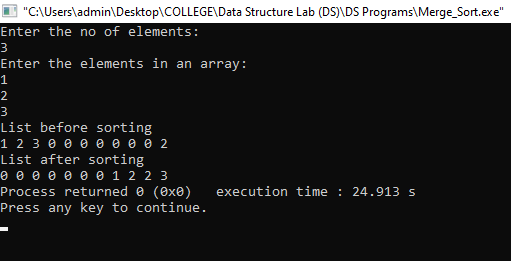
printf("\nList after sorting\n");

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

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

}

**Output:**



**PROGRAM: C program to implement Quick Sort.**

**Input:**

#include<stdio.h>

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 main()

{

int i, pivot, count, a[25];

printf("How many elements you want to enter?: \n");

scanf("%d",&count);

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

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

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

quicksort(a,0,count-1);

printf("Order of Sorted elements: \n");

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

{

printf(" %d\n",a[i]);

}

printf("The pivot element is %d is:",a[pivot]);

return 0;

}

**Output:**

