Program 2

```
#include<stdio.h>
#include<stdlib.h>
#define SIZE 1000
int count;
int partition (int a[], int left, int right)
{
int i, j, pivot, temp;
pivot = a[left];
i = left+1;
j = right;
while(1)
{
while(pivot>=a[i] && i<=right)
{
i++;
count++;
}
while(pivot<a[j] && j>left)
{
j--;
count++;
}
if(i<j)
{
temp = a[i];
a[i] = a[j];
a[j] = temp;
}
else
{
a[left] = a[j];
a[j] = pivot;
return j;
}
}
}
void QuickSort(int a[], int left, int right)
{
int s;
```

```
if(left<right)
{
s = partition(a,left,right);
QuickSort(a,left,s-1);
QuickSort(a,s+1,right);
}
}
int main()
{
int a[100], x[SIZE], y[SIZE], z[SIZE];
int i, j, n, ele, c1, c2, c3;
printf("\nQUICK SORT\n");
printf("\nEnter the number of elements in the array - ");
scanf("%d",&n);
printf("\nEnter the elements of the array - ");
for(i=0;i<n;i++)
{
scanf("%d",&ele);
a[i] = ele;
}
count = 0;
QuickSort(a,0,n-1);
printf("\nThe sorted elements are - ");
for(i=0;i<n;i++)
{
printf("%d ",a[i]);
}
printf("\n\nThe number of counts- %d\n",count);
printf("\nSIZE\tASC\tDESC\tRAND\n"); //for time complexity analysis, using
3

for(i=16;i<550;i=i*2){
for(j=0;j<i;j++){
x[j]=j; //array is filled with elements in strictly ascending order

y[j]=i-j; //array is filled with elements in strictly descending order -->

z[j]=rand()%i; //array is filled with elements in randomn order

}
count = 0;
QuickSort(x,0,i-1); //ascending array is sorted, and number of basic
```

```
c1 = count;
count = 0;
QuickSort(y,0,i-1); //descending array is sorted, and number of basic

c2 = count;
count = 0;
QuickSort(z,0,i-1); //randomn array is sorted, and number of basic
operations

c3 = count;
printf("\n %d\t %d\t %d\t %d",i, c1, c2, c3); //time complexity of merge
sort is

}
printf("\n");
return 0;
}
```