**PROGRAM 9**

**Sort a given set of N integer elements using Quick Sort technique and compute its time taken**

**CODE**

#include<stdio.h>

#include<time.h>

void swap(int \*a,int \*b)

{

int temp=\*a;

\*a=\*b;

\*b=temp;

}

int partition(int a[],int lb,int ub)

{

int pivot=a[lb],i=lb,j=ub;

while(i<j)

{

while(a[i]<=pivot)

i++;

while(a[j]>pivot)

j--;

if(i<j)

swap(&a[i],&a[j]);

}

swap(&a[lb],&a[j]);

return j;

}

void sort(int a[],int lb,int ub)

{

if(lb<ub)

{

int loc=partition(a,lb,ub);

sort(a,lb,loc-1);

sort(a,loc+1,ub);

}

}

void printArray(int arr[],int n)

{

int i;

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

printf("%d\t",arr[i]);

}

int main()

{

int i,n,arr[1000];

double time;

clock\_t start,end;

printf("Enter the total number of elements to be sorted:");

scanf("%d",&n);

printf("The elements to be sorted are:\n");

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

{

arr[i] = rand()%100;

printf("%d\t",arr[i]);

}

printf("\nThe sorted array is:\n");

start=clock();

sort(arr, 0, n-1);

printArray(arr, n);

end=clock();

time = ((double)(end - start))/CLOCKS\_PER\_SEC;

printf("\nTime taken to sort the array by using Quick Sort is:%lf\n",time);

getch();

}

**OUTPUT:**



