LAB-8

QUICK SORT CODE-

```
#include<stdio.h>
#include<time.h>
#include <stdlib.h>
void swap(int* a, int* b)
{
   int t = *a;
   *a = *b;
   *b = t;
}

int partition (int arr[], int low, int high)
{
   int pivot = arr[high];
   int i = (low - 1);
   for (int j = low; j <= high - 1; j++)
   {
      if (arr[j] < pivot)
      {
         i++;
```

```
            swap(&arr[i], &arr[j]);
        }
    }
    swap(&arr[i + 1], &arr[high]);
    return (i + 1);
}

void quickSort(int arr[], int low, int high)
{
    if (low < high)
    {
        int pi = partition(arr, low, high);
        quickSort(arr, low, pi - 1);
        quickSort(arr, pi + 1, high);
    }
}

void printArray(int arr[], int size)
{
    int i;
    for (i = 0; i < size; i++)
        printf("%d ",arr[i]);
```

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



int main()
{
int i,n,sort;
clock_t start,end;
while(1)
{
printf("Enter the number of the elements\n");
scanf("%d",&n);
if(n==-1)
    break;
int a[n];
for(i=0;i<n;i++)
{
a[i]=rand();
}
start=clock();
quickSort(a, 0, n - 1);
printf("Sorted array:\n");
```

```
    printArray(a, n);
end=clock();
double time_taken=((double)end-start)/CLOCKS_PER_SEC;
printf("\n\n");
printf("Time taken for sorting %d elements is %f sec\n",n,time_taken);
printf("\n");
}
}
```