S.No: 3	Exp. Name: ***Quick sort***	Date:2025-07-14

Aim:

Write a program to perform Quick sort. Display the partial pass-wise sorting done.

Source Code:

quickSort.c

```
#include<stdio.h>
void displayPass(int arr[], int l ,int r)
{
   printf("Pass: ");
   for(int i = l; i<= r; i++)
      {
         printf("%d ",arr[i]);
      }
   printf("\n");
}
void swap(int *a, int *b)
{
   int t = *a;
   *a =*b;
   *b = t;
}
int partition(int arr[20], 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);
      displayPass(arr,low,high);
      quickSort(arr,low,pi -1);
      quickSort(arr,pi+1,high);
   }
}
int main()
{
   int n;
   printf("number of elements: ");
   scanf("%d",&n);
   int arr[n];
   printf("elements: ");
   for(int i = 0 ; i< n;i++)
      {
         scanf("%d",&arr[i]);
      }
   printf("Original array: ");
   for(int i = 0 ; i< n; i++)
      {
         printf("%d ",arr[i]);
      }
printf("\n");
quickSort(arr,0,n-1);
printf("Sorted array: ");
for(int i = 0; i< n;i++)
   printf("%d ",arr[i]);
printf("\n");
return 0;
}
```

Execution Results - All test cases have succeeded!

Test Case - 1

User Output

```
number of elements:  4
elements:  5 8 9 4
Original array: 5 8 9 4
Pass: 4 8 9 5
Pass: 5 9 8
Pass: 8 9
Sorted array: 4 5 8 9
```

Test Case - 2

User Output

```
number of elements:  6
elements:  5 1 10 8 9 7
Original array: 5 1 10 8 9 7
Pass: 5 1 7 8 9 10
Pass: 1 5
Pass: 8 9 10
Pass: 8 9
Sorted array: 1 5 7 8 9 10
```