

Aim:

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

Source Code:**quickSort.c**

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

void pass(int a[],int low, int high){
    for(int i=low;i<=high;i++){
        printf("%d ",a[i]);
    }
    printf("\n");
}

int Partition(int a[],int low, int high){
    int pivot=a[high];
    int i=low-1;
    for(int j=low;j<high;j++){
        if(a[j]<pivot){
            i++;
            swap(&a[i],&a[j]);
        }
    }
    swap(&a[i+1],&a[high]);
    return i+1;
}

void quickSort(int a[] ,int low, int high,int n){
    if(low<high){
        int pivot=Partition(a,low,high);
        printf("Pass: ");
        pass(a,low,high);
        quickSort(a,low,pivot-1,n);
        quickSort(a,pivot+1,high,n);
    }
}

int main(){
    int n,a[10];
    printf("number of elements: ");
    scanf("%d",&n);

    printf("elements: ");
    for(int i=0;i<n;i++){
        scanf("%d",&a[i]);
    }
}
```

```

printf("Original array: ");
for(int i=0;i<n;i++){
    printf("%d ",a[i]);
}
printf("\n");
quickSort(a,0,n-1,n);
printf("Sorted array: ");
for(int i=0;i<n;i++){
    printf("%d ",a[i]);
}
printf("\n");
}

```

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