Practical 5

```
#function that consider last element as pivot,
#place the pivot at its exact position, and place
#smaller elements to left of pivot and greater
#elements to right of pivot.

def partition (a, start, end):
    i = (start - 1)
    pivot = a[end] # pivot element

    for j in range(start, end):
        # If current element is smaller than or equal to the pivot
        if (a[j] <= pivot):
            i = i + 1
            a[i], a[j] = a[j], a[i]

    a[i+1], a[end] = a[end], a[i+1]

    return (i + 1)

# function to implement quick sort
def quick(a, start, end): # a[] = array to be sorted, start = Starting index, end =
Ending index
    if (start < end):
        p = partition(a, start, end) # p is partitioning index
        quick(a, start, p - 1)
        quick(a, p + 1, end)


def printArr(a): # function to print the array
    for i in range(len(a)):
        print (a[i], end = " ")


a = [68, 13, 1, 49, 58]
print("Before sorting array elements are - ")
printArr(a)
quick(a, 0, len(a)-1)
print("\nAfter sorting array elements are - ")
printArr(a)
```

Output:

```
Before sorting array elements are -
68 13 1 49 58
After sorting array elements are -
1 13 49 58 68
```